



Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

**GIAC Certified Windows Security Administrator (GCWN)
Practical Assignment Version 3.1 (revised April 2002)
Option 3**

Securing Web Based Application Service Provider - Case Study.

Alexander Tarasul

March 2003

© SANS Institute 2003, Author retains full rights.

Table of Content

Table of Content	2
Abstract.....	3
Introduction	4
The business case	4
Market segment.....	4
System architecture overview	4
Securing data in transit.....	5
Backend server functionality and network requirements	6
Risk Structure.....	7
Assets.....	7
Threats	8
Risks.....	8
Summary	9
Defense in depth	10
Physical security / Hosting ISP layer	10
Local firewall layer	11
IPSEC layer	12
OS security layer	12
Application layer	13
Attack detection.....	13
Users/Human Factor/Internal Security Risks.....	14
Building Bastion Host (Windows and IIS security)	14
Site architecture – set of standalone servers	14
Consistent installation process using slipstreamed service pack CD and answer file.	16
Build installation CD.....	16
Use unattended installation for consistency.....	16
Hardware and partitions.....	17
Post Installation Steps	17
Building security template for consistent hardening configuration.....	18
Select source template	18
Parsing templates using MSSQL	19
Template settings for AlsAsp.....	20
Event Logs.....	20
Password Policy	20
Account Lockout Policy.....	21
Audit Policy.....	21
User rights assignment and restricted groups	21
Security options	25
Custom Registry Settings	26
Event Log	26

System Services	26
Registry and File System hardening	29
IIS Security	29
IIS Lockdown tool	30
SMTP Security.....	31
PCANYWHERE – Securing Remote access	32
Access Control/Authentication.....	32
Confidentiality	33
Integrity	33
Auditing.....	34
Identified security problems	34
MSSQL Security	34
Issues and Goals.....	34
SQL Server Installation configuration.....	35
Connections and user authentication	36
Note about back-runner - SQL Mixed (Native) authentication	37
Notification of administrators and auditing.....	38
Database permissions.....	39
Access to AlsData database.....	39
Access to other server databases.....	40
Results of SQL Server hardening.....	40
Application Security	40
Web application security.....	40
Other applications security	42
Operations Security	42
Problem recovery and business continuity.....	42
Backup procedures	43
Patch management.....	43
Security QA – penetration testing.....	44
REFERENCES	45
Appendix 1	45

Abstract

This paper presents security architecture and implementation for small application service provider, running in co-hosted environment. The paper highlights all aspects of information security, including network, OS, physical, business continuity, disaster recovery, remote administration etc. The paper also describes steps to secure IIS, web application, and MSSQL Server as well as custom application used by ASP. Most of measures and decisions included into architecture have consensus acceptance and wide coverage in white papers and books. The author carefully mentioned them, but not covered in deep detail. This allowed devoting most of attention to issues, which security papers barely cover, such as using PCAnywhere for remote administration, password policy for web applications, comparing security templates etc.

Introduction

The business case

AlsAsp.com is a small privately held startup company in a business of application service provider. Alsasp.com specializes in vehicle tracking and automatic vehicle location. The business advantage of the AlsAsp is in integrating wireless GPS devices, wireless data plans, and mounting hardware and reliable web based client software into attractively priced packages and selling it to corporate customers. AlsAsp operates on extremely small margins and to remain successful it has to be very cost-effective. At the same time to stay in business, it wants to address efficiently all security risks it faces.

Market segment

In our time of multiple security guides and architectures, it seems like there is a large number of businesses, which fall under radar of security architect. Those are small companies, which host their own Internet solutions on either owned or more frequently in co-hosted environment. They run fully featured applications open to their clients over the Internet. While functionally they offer is close to interned data-center, financial constraints and different risk structure do not allow them to be secured using corporate interned data-center checklists and required expensive resources. Usually they cannot afford specially tailored to them security architecture.

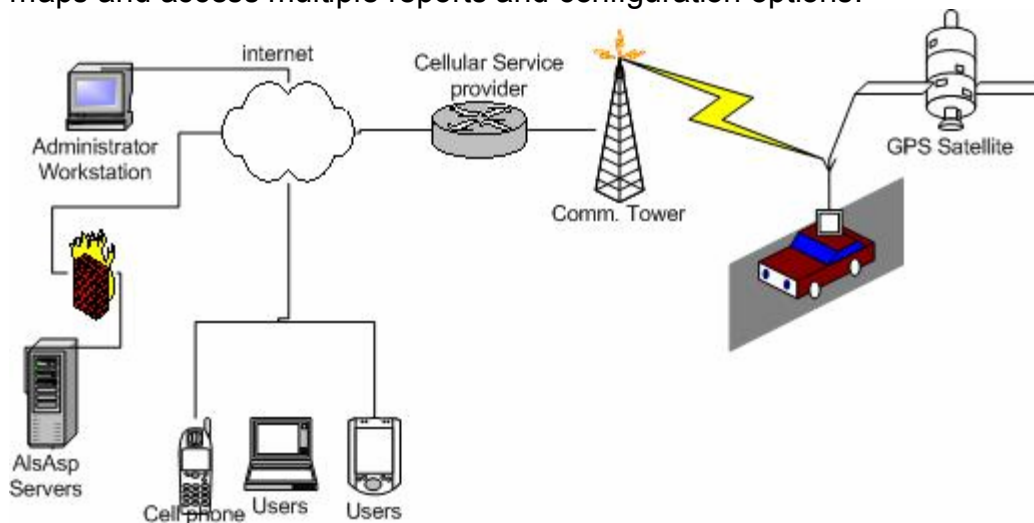
However, they are much different and have more resources then Internet exposed SOHO and should be protected much more intensively then described in guides oriented on home user. They constitute significant part of Internet, their security failures have both significant monetary value and provide ground for DDoS attacks, and penetration hops on bigger target. Their security solution should not be limited to installing purely configured firewall and sporadic patch installation.

This paper describes security architecture and security operations for small-size Application Service Provider. This is a subclass of co-hosted solutions, providing Internet delivered services and not accepting payments for them over the Internet. Author used his own experience in securing businesses very close to described here as well as several published guides from Microsoft and NSA.

System architecture overview

AlsAsp deployed wireless [GPS](#) devices on client vehicles and registered those devices with cellular service providers (different providers depending on location, etc). When powered those devices automatically register with wireless network and report their location on pre-programmed scheduled basis. The location packet containing timestamp, device ID, source IP address, location, speed and direction is binary packed and delivered through cellular network to cellular service provider and then repackaged as UDP packet through Internet to pre-

programmed destination IP address of AlsAsp application server. UDP listeners (custom written Windows 2000 multithreaded service) unpack binary packed information and submit to MS SQL Server database. The scheduled process in database processes queue of messages and then reverse geocode it. Users can log-on into web-based application (Active Server Pages connected to SQL Server), view location of their vehicles using real-time generated graphical maps and access multiple reports and configuration options.



Initial architecture design assumed that single server would host functionality of application, database and web server, and run in co-hosted environment with ISP. All other company operations (sales, support etc) are in office remotely located from production environment. This paper describes securing production environment only. For the purposes of this paper, the company office is just a location of one of two remote administration workstations. Another administrative endpoint located in administrator home office.

Securing data in transit

Vehicle location data is very important and AlsAsp takes all reasonable measures to protect it. By the nature of solution architecture, it is possible to intercept this data in transit before it reaches AlsAsp servers or between servers and user's browser. Determined attacker can intercept it on the way from GPS device to cellular service provider and from cellular service provider to AlsAsp. This level of exposure is acceptable for current customers and is very cost-effective for integration with multiple geographically dispersed providers. It is possible, however, on customer demand to establish much more secure transit here. First of all CDPD protocol used for most of AlsAsp devices is pretty well encrypted and authenticated (see [CDPD Security](#)) and the segment from device to cellular provider can be considered well protected. The company considers internal path of data packet through cellular provider is secure through the proper use of Service Level Agreement. Finally, it is possible for most of cellular

providers to deliver data packets to AlsAsp using IPSEC or VPN channel between provider and AlsAsp (for additional charge).

The data is also exposed on transit from server to client web browsers. All current clients accept risk of HTTP data intercept. AlsAsp currently protects using SSL only form based authentication exchange, which includes user login and password. All subsequent data traffic to authenticated users is unencrypted for performance reasons and identified by session cookie. AlsAsp is ready to deploy for security conscious clients the web site completely protected by SSL with client side certificate pre-authentication – the connection will be possible only with AlsAsp issued certificate presented by client computer, but final authentication still will be made by login/password authentication. All HTTP traffic between browser and server will be encrypted using HTTPS.

Backend server functionality and network requirements

AlsAsp designed the backend functionality to run on single hosted server. To analyze security requirements and prepare application to scale horizontally the company has identified the separate “building blocks” of functionality and their network visibility. It is critical for server security to know exact minimum set of required network connections (incoming and outgoing) and have those and only those connections allowed through firewall and IPSEC layers.

The production web site is running on IIS server and is visible on ports TCP 80, 443 to the world. It has outgoing connection to MSSQL database server if it running on other machine. QA web presentation is visible on ports 80,443 to selected addresses. It has outgoing connection to MSSQL database server if on other machine.

MSSQL Database Server is visible from correspondent web site on port 1433 through IPSEC ESP if it is running on separate machine otherwise not visible over TCP at all – connection happens through shared memory protocol.

The backup web server is not visible to the world. Actually, it powered off all the time except when administrator upgrades it.

The web site for internal file transfer is running on port 88 and visible only to selected addresses – other hosts on the site.

The application server, which accepts incoming vehicle location data, is visible on UDP ports 3010-3020 to the world and needs outgoing connection to MSSQL database server if on other machine.

Remote administration is visible on PCAnywhere ports – 5631 TCP and 5632 UDP only to selected addresses.

Outgoing email requires access to port 25 and DNS port 53 (TCP/UDP).

To provide business functionality AlsAsp should configure servers to synchronize automatically the system time with reference clocks. This was done using public Internet timeservers, SNTP protocol and Windows Time service.

From <http://www.boulder.nist.gov/timefreq/service/time-servers.html> AlsAsp choose three primary SNTP servers and run following configuration command on each server:

net time /setsntp:"129.6.15.28 129.6.15.29 132.163.4.101"

The windows Time service on regular basis contact those servers and adjust computer time. The synchronization by default happens once every 45 minutes until three good synchronizations occur, then once every 8 hours.

To function this service need outgoing access from port 123 to port 123 on those three destination servers.

Another outgoing access to Internet is required for patch management. The patch management section describes that servers are running hfnetchk.exe as a scheduled task. This program downloads fresh XML file with patch information from either Microsoft or Shavlik and compare current server files and registry against information in this XML file. To run this program the outgoing connection required to <http://www.shavlik.com> or <http://downloads.microsoft.com>. Those servers also have high chance to change IP address and therefore require DNS and not IP address naming in firewall and IPSEC filters.

AlsAsp solved this problem in following way. The offsite administrator's workstation has unrestricted outgoing access to Internet and run hfnetchk.exe daily as a scheduled task. After the run mssecure.cab, which contains XML file is on file system of administrator's workstation. The folder on administrator's workstation is exposed as web site on port 88 with limited visibility – visible only to IP subnet of AlsAsp servers. The servers, when run hfnetchk.exe, specify parameter "-x <http://123.123.123.123:88/mssecure.cab>", and force download of fresh XML file from administrator's workstation.

In this way, the outgoing connection requirement, required to be open on firewall and IPSEC, limited to one fixed port on one known computer. The same time the functionality of automatic patch checking is preserved without compromise.

AlsAsp is using the same website on administrative workstation as staging area for patch downloads.

The careful minimizing of incoming and outgoing connection significantly limits potential attacker abilities. There is no way, for example, to fire back shell to the host of attacker choice or execute cross site scripting attack on the server, as firewall allows outgoing connections only to few selected ports on few selected hosts.

Risk Structure

While the best solution for securing AlsAsp would be to apply enterprise level guidelines, the financial constraints have made this impossible. The company has analyzed assets to protect possible threats to them and risk structure.

Assets

The critical assets AlsAsp wants to protect are (in order of decreasing priority):

1. Company reputation
2. Backend source code
3. Work data (vehicle locations archive)

The reasons behind this prioritization are below.

Backend source code has additional legal protection through copyright and insurance clauses, while company's reputation surely does not enjoy such protections.

Threats

AlsAsp categorized possible threats based on Microsoft [STRIDE](#) model (See Stride Model), which identifies threats as spoofing identity, tampering with data, repudiation, information disclosure, denial of service, elevation of privilege.

Technical measures to protect user authentication information (see Web Application Security) mitigate "Spoofing identity" threat.

"Denial of service", "Tampering with data", "Information disclosure" and "Elevation of privilege" threats are mitigated by technical measures to protect Windows 2000 Server, IIS and SQL Server.

Detailed application level logging and web server logging mitigate the "Repudiation" threat.

The critical unmitigated risk is administrator error. In multi-user corporate environment it is possible to reduce this risk by separating and delegating administrative authority. AlsAsp has only one administrator and as such has to assume this risk.

Errors and omissions insurance clause provides additional mitigation.

Risks

The company has listed and prioritized major security risks and analyzed what it needs to do to address them.

1. The service interruption risk is the most critical for the company. The service can be interrupted by attacker, who gained root access to the servers. Many technical measures described below have a goal to prevent it.

Without login to the server attacker can attempt to interrupt service by triggering Denial of Service attack. The relatively cheap firewall (see local firewall layer section) can only partially address brute force DoS to open production ports (mostly HTTP/80). AlsAsp has agreement with hosting ISP and 24x7 ISP service is ready to respond on requests to block access from attacking IP address using powerful ISP border firewalls/routers. There is no way, however for AlsAsp to survive targeted and determined DDoS attack. This risk is accepted and partially mitigated by keeping low profile.

The company experience and multiple publications suggest that internal sources can be more likely source of service interruption than external attacker.

Those sources identified as hardware and software failures as well as administrator errors.

Business Continuity Plan addresses hardware failures and critical software problems. Application failures addressed by rigorous QA process, usage of QA

server and Change Management plan. Administrator's education should reduce risk of administrator errors.

2. The risk of source code theft is significant for the company. The attacker with root access to the server definitely can obtain almost all source code. There is no technical protection for JavaScript/HTML code, downloaded to user browser. To minimize losses AlsAsp took following measures:

- Keep the front-end code as simple as possible – the bulk of proprietary logic is on better-protected backend.
- The source is not available for casual viewer by disabling right clicks on web page and removing browser's menu bar.
- To get access to client side code the attacker need to login into application with valid password – user password protection implemented to reduce this risk.
- AlsAsp did considered usage of [Microsoft Script Encoder](http://www.klaphek.nl/nr6/scrdec.html) - (screnc.exe), but availability of tool (<http://www.klaphek.nl/nr6/scrdec.html>), which breaks this encoding, has made this protection meaningless.
- AlsAsp put most of JavaScript code into separate JS files and enabled IIS compression on files with *.JS extension by changing metabase property. This change prevents casual viewing of file in transit and optimizes loading performance, but definitely does not stop determined attacker.

3. Usage of company servers to plant attacks on other servers and associated liability issues can cause significant problem. Multiple security measures protect server from attacker gaining any OS level user access to computer. Errors and omissions insurance partially mitigate this risk too.

4. The modification of customer data is very difficult for attacker to execute, but still treated as significant risk by AlsAsp. Modification of data carried by UDP packet in transit requires significant knowledge to keep binary packed data well formed and recognizable. The cellular network segment is well protected by encryption. The segment from cellular provider to application server usually contain small number of hops going through highly protected routers, as both cellular providers and AlsAsp locate their computers close to major backbone – this is one of the places where co-hosting choice is strategic. Layers of OS and SQL Server security protect the data on database server. This data is highly de-normalized and at the same time is partially redundant. Significant knowledge of data model is required to modify data without ability to detect modification.

5. Web application security mitigates this risk of unauthorized user access to application and viewing of customer data.

Summary

The main goal of AlsAsp security architecture is to provide uninterrupted service to clients. The company needs to provide due diligence in protecting backend server OS and application from unauthorized OS level access. OS and applications should be hardened using their existing security features.

The change management process should ensure backend server stability during changes in both purchased and in-house software. Application design should make as difficult as possible to spoof user identity by guessing user credentials. Carefully thought over business continuity process must allow company to recover from any failure/security breach as soon as possible. The goals should be achieved without major investments to security specific products and services. Instead, the capabilities of existing software and hardware should be used in full extent. Not addressed risks should be transferred to insurance and legal protection.

Defense in depth

As said in "Defense in Depth" by Brian McKenney¹⁰, the defense in depth strategy combines the capabilities of people, operations, and security technologies to establish multiple layers of protection. An intruder must circumvent these defenses in order to gain unauthorized access. With defense in depth, the objective is to implement defenses at multiple locations to protect critical resources so they can continue to operate in the event that attacker circumvented one or more defenses. While planning defense for each layer, one layer should not rely on presence of other layers. The failure or penetration of one defense layer should not affect another layer. The goal is to have two or more defenses for each of possible attack directions. AlsAsp implemented defense in depth using multiple layers described below.

Physical security / Hosting ISP layer

Hosting ISP provides outermost layer of network security, as well as full physical security. This layer provides defense of some unauthorized network access and it is the only layer, which provides defense from unauthorized physical access and environment failures. This includes:

- ISP configured external firewall on principle of "ALLOW ALL / DENY SELECTED". While this firewall keeps most of the ports open (to minimize number of rules different per hosted company) it do plays important role. ISP has closed the critical ports - TCP 139,445, the ICMP pings are disabled. On short notice, the list of closed critical ports can be expanded (UDP 1434 was closed early in recent Slammer attack).
- Three independent uplink connections to different backbone providers guarantee high uptime.
- DOS attack blocking (on request) allows transferring DOS load to more powerful ISP routers.
- Separate switched subnet for the rack stand prevent neighborhood sniffing (it is still possible to do ARP poisoning)
- Physical access security – the live guard, carded door and locked rack stand.
- HVAC security provided by two independent climate devices
- Power conditioning and one hour data center wide ups generator

- Easy reachable in less then 30 minutes location

Local firewall layer

AlsAsp is using relatively inexpensive firewall NetScreen 5XT firewall. This firewall limited in number of connection and performance, but provides all the features of expensive big scale NetScreen firewalls. The growth plan assumes that when company outgrows this firewall the network will need to segmentation. In this moment the company will push NetScreen 5XT back, separating database and infrastructure servers from front-end servers, while newer, more powerful firewall will take its place.

Firewall configured in transparent mode as Layer 2 IP security bridge and fine tuned by “DENY ALL/ALLOW SELECTED” principle, opening only protocols and ports required for company functioning. Except opening only selected ports, NetScreen firewall provides multiple defense features. Partial list of features used by AlsAsp includes:

- DoS protections against SYN attack, ICMP flood, Ping of Death, Tear-drop, ICMP Fragment, Large Size ICMP Packet, SYN-ACK-ACK attacks.
- WinNuke and Land Attack Protection
- Malicious URL filtering patterns (EXE, DLL) Firewall provides packet reassembly to protect against malicious URL split between packets. It can filter URLs based on patters
- Source and Destination IP Based Session Limit set very high (about 75%) of firewall capacity to give a chance to survive against DOS attack.

While firewall can terminate VPN, AlsAsp has chosen to traverse administrative IPSEC connection through it and authenticate on the IPSEC layer.

Administrator can manage firewall only from trusted interface (from company servers) – external administration connection prohibited. Firewall authenticates user connection by using internal user database with login and password different from other access passwords.

NetScreen firewall is a powerful protection layer. However, this is a separate piece of hardware/software with its own vulnerabilities. AlsAsp carefully monitors vulnerability alerts for NetScreen by subscribing on vendor security alerts bulletin and includes NetScreen updates into patch management plan.

With all its advantages, the firewall poses significant assumed threat to the company operations. Should attacker manage to control firewall or administrator incorrectly configured it or firewall just simply fail then the DoS will occur.

Administrative connection will not be possible, as it first need to cross the same failed firewall. The current mitigating plan is to drive to data center and reload configuration from saved file or reconnect network, bypassing firewall.

While the NetScreen firewall is highly reliable for site growth plans, AlsAsp is considering higher levels firewalls (from NetScreen 50 and up), which should provide better performance together with high availability clustering. At this stage, the site segmentation will increase and current firewall will take place between web and database servers.

IPSEC layer

The IPSEC provides major defense layer for an AlsAsp with main emphasis made on packet filtering functionality of IPSEC configured as DENY ALL/ALLOW SELECTED. The filtering rules for IPSEC policy are mostly identical to firewall filtering rules. In addition to firewall filtering rules, they also permit communication between hosts behind firewall. IPSEC allow through the following incoming ports:

TCP 80 from the world	Primary Web server functionality
TCP 443 from the world	SSL web server
TCP 25 from mainstream mail server (see SMTP security below)	Mail exchange
TCP 53, UDP 53 from upstream DNS servers	DNS
UDP 3010-3020 from the world	Application functionality – incoming location data
TCP 5631, UDP 5632 from 2 administrative workstation	PCAnywhere remote administration
TCP 1433 from sister hosts	MSSQL Server
TCP 88 from sister hosts	Internal file transfer

While the IPSEC layer mostly duplicates firewall and adds additional layer into defense in depth strategy, it does not replace firewall functionality. Broadcast, multicast and IKE traffic are exempt from IPSEC filtering even with Nodefaultexempt registry setting set. It is possible to mitigate this problem by using the IPSEC layer together with “TCP/IP filtering” functionality on network adapter. This functionality filters only internal traffic, can not make decisions based on source address and does not filter ICMP. AlsAsp configured “TCP/IP filtering” to allow TCP ports 80,443,1433,5631,25,53 and UDP ports 3010-3020 5632, 53 and protocols 6 and 17 (TCP and UDP) only. In this way combined IPSEC and “TCP/IP filtering” layers completely filter out incoming broadcast, multicast, IKE and ICMP traffic.

Other shortcomings of IPSEC that make it less functional then firewall:

- IPSEC does not track state of connection - there is no way to find out if connection was initiated from inside or from outside
- IPSEC does not allow specifying range of ports.
- IPSEC does not have logging facility to review rejected packets.

In case of AlsAsp, those shortcomings mostly affect convenience of administrator, who needs to create multiple rules, where one rule with port range would otherwise work.

OS security layer

This layer works for valid incoming connections to keep them secured with proper ACLs permissions. It provides security for ports which server listens to, but do not allowed through firewall and IPSEC layers – accomplishing defense in

depth strategy. Those ports are TCP 135, 445, UDP 445 for RPC service. OS security also should ensure that in case of buffer overflow attack, when attacker assumes identity of low privileged account, the attacker will be kept with those low privileges without ability to escalate them. See Template Settings section on actual setting for OS Security.

Application layer

Safe coding practices and secure application policies help restrict access to application to authorized users and keep application data safe. In addition, by not allowing unauthorized user to login into web application, this user is much more limited to the ways it can penetrate server security. See Application security section below for more details.

Attack detection

While most of this paper devoted to attack prevention, we should also note that attack detection plays very important role. One very simple and inexpensive ways to detect problem is to run or subscribe to monitoring service. AlsAsp subscribed to several monitoring services (<http://www.securityspace.com> is one of them), which periodically (from 5 to 15 minutes depending on service) attempt to retrieve specific web page from the server. This web page in turn connects to database and create required COM component. If something went wrong (no connection, database failure, server busy etc) the page will not return predefined HTML to monitoring service and service will escalate problem by emailing to pagers/phones of administrators. By subscribing to several independent services the problem of accidental service maintenance during AlsAsp problem eliminated. This monitoring proved very effective and runs with very low false alarm rate.

AlsAsp is currently missing effective, reliable and affordable host based IDS and anti-virus system. The major problems here are:

1. Acquisition cost – there is significant gap between \$30-\$40 personal firewalls and \$700 and above for enterprise grade products.
2. Support cost – host based IDS issue many false positives on recommended configuration level. Sorting through them is cost prohibitive for a company without full-time stuff allocated to this – what really required here is 24x7-support desk. Lowering configuration level (usual reaction of users) is impossible
3. Effectiveness of notification – without a separate channel for IDS alert it is naïve to assume that in case of determined attack the IDS alert will be allowed by attacker to be delivered to administrator by email – attacker already controls internet connection. Allowing IDS to react automatically (block addresses etc), will pose significant DoS risk.
4. IDS and anti-viruses require administrator or service/local system account to run. Highly privileged piece of software with its own bugs, vulnerabilities and exploits may do more harm then good.

AlsAsp, however, is researching several freeware products, such as Snort (<http://www.snort.org>) and AVG 6.0 Antivirus (<http://www.grisoft.com>), as potential solution for IDS/anti-virus defense layer.

Users/Human Factor/Internal Security Risks

While security researchers point out that 80% security threats is internal this risks are significantly lower for AlsAsp. Only two peoples have administrative access to servers. Only primary/security administrator knows regularly changed complex password to renamed administrator account. Second administrator (company owner) has separate administrator account. Both administrators are not susceptible to social engineering attacks and do not write down password. The only vector of attack here is gaining full remote control of administrative workstation with key logger installed to record all passwords. AlsAsp understands and accepts this risk. While this is outside the scope of paper, the administrative workstation are fully patched Windows 2000 workstations, running firewall/anti-virus applications, secured with templates very close to those used to secure servers and IPSEC enabled. No other company employee has access to administrative workstation and to hosted servers.

Another vector of attack is social engineering means of obtaining application access credentials by unauthorized web application users. AlsAsp established the procedure of application password reset. User can call help desk to reset password. User needs to provide his company name, username, full name, and answer to security question. If this information matches to database, the help desk will reset the password and email notification with new password to the registered client contact - not to email provided by user.

Other user accounts on servers are services accounts. The passwords for those accounts satisfy password policy and set by administrator on each server individually. The administrator does not keep record of service account passwords after they set on user account and service manager. In this way administrator has no way to accidentally disclose those passwords.

Building Bastion Host (Windows and IIS security)

Site architecture – set of standalone servers

AlsAsp has chosen to run the server farm as a set of standalone servers. The root of this decision is that AlsAsp hosting site historically started from one Windows NT server and there were absolutely no advantages to create NT Domain. As site grew to several Windows 2000 server, AlsAsp revisited this decision again. The following arguments were weighted against converting to AD domain:

- AlsAsp is running very simple infrastructure – no interactive users except 2 Administrators and few service accounts, one logical OU, one logical site.
- For security reasons bastion hosts are running the bare minimum set of services. AD infrastructure would add many services only to support infrastructure running.
- The communications between computers are highly limited. AD infrastructure will require opening additional ports and IPSEC secure/encrypt communications.
- Hardware overhead – at least one additional computer would be required as well additional firewall to segment site. This will not only require initial investment, but will incur significant recurring cost by adding to hosting rack space requirements.
- Administrator overhead - part-time administrator would need to spend time on maintaining AD infrastructure and configure infrastructure server differently from web, application and database servers.

AlsAsp also weighted AD advantages:

- Group policy will allow consistent application of configuration changes to all computers. This advantage is not significant for AlsAsp, which achieves the same results in compatible time by batch file application of templates using secedit.exe (see below).
- Possible acquisition of customers with heightened security requirements might require client certificate authentication. The PKI maintenance will be advantageous using Active Directory.
- Kerberos authentication of domain accounts will allow more transparent authentication between services running on different computers.

The later two advantages are the main for AlsAsp plans to consider move to Active Directory domain. The projected milestone to review advantages and disadvantages of Active Directory is on 2 times hardware growth stage – reaching 8 servers on site.

While future windows version will provide standalone Active Directory Service, it will not satisfy domain infrastructure needs.

Consistent installation process using slipstreamed service pack CD and answer file.

Build installation CD

The procedure of building bastion host in AlsAsp started from installation. AlsAsp is using slipstreamed Windows 2000 Server SP3 (as of latest) bootable CD with custom answer file. The procedure of creating such CD described at <http://www.bink.nu/Bootcd/default.htm> and requires original Windows 2000 server CD to be copied into hard drive folder, service pack extracted to another folder using command "w2ksp3.exe -x" and service pack slipstreamed into windows installation using D:\win2kSP3\i386\UPDATE\UPDATE.EXE -S:D:\CD-root.

The administrator should burn CD with resulting installation image. As only i386 installation folder needed for installation then remaining space on resulting CD can be taken by other useful installations (AlsAsp is putting here all windows hardening tools and scripts and full (more than 70 Mb) installation of Internet Explorer 6 SP1). This helps to achieve important task of not connecting computer to the Internet before hardening completed – default Internet Explorer installation is small file that demand Internet connection for downloading actual installation binaries.

We were not able however to create bootable using Nero and advice on this page. Another resource - <http://www.nu2.nu/bcd/> - allowed us reliably generates bootable installation CD.

Administrator periodically updates CD by burning in intermediate hot-fixes and updating administration scripts. It is mandatory updated on each major OS Service Pack. Having as fresh CD as possible allows minimizing potential recovery time in case of emergency server build.

Use unattended installation for consistency

AlsAsp is using partially unattended installation in mode HideDefault.

Administrator created installation answer file winnt.sif (see Appendix 1) using Windows Setup Manager Wizard (see Microsoft KB Article [250609](#)) and placed on the floppy.

When setup is running in HideDefault mode, it prompts user for not entered data such as administrator password, but does not show selections and options defined in winnt.sif file. This allows for making consistent installation with only required features enabled, such as IIS installed without samples, only required network components selected, etc.

Hardware and partitions

AlsAsp servers are rack-mounted dual-processor servers DELL PowerEdge 2550-2650. All of them utilize from three to four hot-pluggable 36 GB SCSI drives in RAID-5 configuration. On servers with four disks, one disk configured as hot spare. All of them have a pair of Intel Pro 100 Secure Server NIC NICs. This allows the server to utilize IPSEC for encryption of traffic without a significant performance hit to the CPU.

Installation process prompts administrator to manually partition hard drive (setting AutoPartition=0) as there is no way to specify default partitioning of hard drive in winnt.sif. The RAID-5 drive partition should be created using OEM tools in advance. Administrator should create following partitions:

Name	Minimum size	Purpose
SYSTEM	4 Gb	Windows 2000 OS
APPLICATIONS	4 Gb	Custom Applications
CONTENT	4 GB	Web Server Content
DATA1	8 GB	SQL Server Data
DATA2	8 GB	SQL Server Data
WORK	8 GB	Logs, Backups, tools, etc

Installation formats system partition with NTFS file system.

Post Installation Steps

- Run post installation batch file postinstall.bat. The major tasks of this batch file are to format all partitions with NTFS, label them, apply initial set of templates (see template section below for details), and create service users
- Install MSSQL Server (see MSSQL Security for details) with latest server pack, AlsAsp custom content, applications, and all patches available on CD. The server will not be connected to the network until all latest critical security patches will be applied (it is possible however to install non-critical patches later by normal installation process)
- Restore IPSEC configuration from backup stored on installation CD and assign AlsAsp IPSEC policy
- Move default website to content folder on content partition
- Configure IIS logging to on work partition
- Run IIS lockdown wizard in Active Server Pages mode
- Configure URLScan (see IIS security layer for details)
- Apply final set of templates (most of the services will be disabled)
- Reboot computer
- Copy latest database backups to work partition and run database restore scripts

- Delete all installation related scripts from computer
- Reboot and connect to network.
- Run hfnetchk to verify that all patches have been applied correctly.
- Test application functionality

Building security template for consistent hardening configuration

Select source template

The security templates play critical role in securing AlsAsp servers. The company chooses the following strategy with security templates – select best available security template, periodically update it based on new requirements and finding, split it on two parts – one to apply just after installation and another to apply periodically to keep settings consistent.

AlsAsp selected following templates for initial review:

Template	Name	Source
Securews	Secure Workstation	Original Windows installation
hisecls	Increases Securews Settings	Original Windows installation
basicsv	Default Security Settings	Original Windows installation+SP3
Hiseclsweb	High security web server	Microsoft
FMStocksPresentation	Sample template	The Security Operations Guide for Windows 2000 Server
w2k_server	Windows 2000 server	NSA Security Recommendation Guides
CIS-Win2K-Level-I-v1.1.7	Window 2000 server level 1	Center For Internet Security https://www.cisecurity.org/
Baseline	Windows 2000 baseline security	Microsoft / eWeek Open Hack 4
cc_server	Windows 2000 Common Criteria	Microsoft Windows 2000 Common Criteria Target
cc_server_sec	High Security Windows 2000 Common Criteria	Microsoft Windows 2000 Common Criteria Target

There was not any available source, which described solution to the task of finding “best available” template. Usual recommendation is to select template from source you trust and then customize this template for company needs. In

this case, the administrator picked up 10 templates from six publications from three trusted sources.

While Microsoft has functionality in Security configuration tool, which allows for comparison between current computer setting and template it does not provide functionality to compare two templates between each other or merge them. Let alone there is no functionality to compare multiple templates.

The article "Securing an IIS 5.0 Web Server on Windows 2000 using Security Tools and Templates" at http://www.sans.org/rr/win2000/sec_IIS5.php describes Perl scripts to compare two templates.

The lack of the tool, which would allow comparing templates pushes administrator to select template based on guess.

Parsing templates using MSSQL

To overcome this problem AlsAsp has used the tool, which allows very flexible analysis of this data -MSSQL Server. Attached script (Appendix 1) shows how to load template data into SQL Server. The templates text files loaded to buffer table using bcp.exe (bulk data transfer) and then parsed into relational table structure.

While most of the settings have obvious name for administrator it is possible to consult sceregvl.inf (included into NSA package) to decode values into human readable settings we can see in security configuration tool. Sceregvl.inf provides way to create your very own mappings of custom settings to interface. While convenient, it is not required - one can enter setting into text template without it appearing in security configuration interface. This feature also has no effect on deployment. The future enhancement of template parsing script will allow parsing sceregvl.inf into relational table.

Having template data in relational table allow to issue queries that fetch different values for the same settings from all templates. Here is, for example, the query to compare values of "Maximum Security Log Size"

```
select template,keyname,value from infs  
where sectionname='[Security Log]' and keyname='MaximumLogSize'  
order by value,template
```

baseline	10240
cc_server_sec	10240
FMStocksP	10240
Hisecweb	10240
hisecws	10240
OHBaseLine	19968
w2k_server	4194240
basicsv	512
Securews	5120

Using the information above AlsAsp has made informed choice of this value as 19968.

Template settings for AlsAsp

While whole template used by AlsAsp available for download (see Appendix 1) the reasoning behind most critical settings described below.

Event Logs

AlsAsp has disabled “Event Log/Shut down the computer when the security audit log is full” to warrant against DoS – the availability is the first priority. The 20Mb is a reasonable size of event log. 15 days is an adequate retention time to backup log. Weekly backups of the log allow for reliable audit trail. The guest access to logs is disabled.

Policy	Computer Setting
Maximum application log size	19968 kilobytes
Maximum security log size	19968 kilobytes
Maximum system log size	19968 kilobytes
Restrict guest access to application log	Enabled
Restrict guest access to security log	Enabled
Restrict guest access to system log	Enabled
Retain application log	15 days
Retain security log	15 days
Retain system log	15 days
Retention method for application log	By days
Retention method for security log	By days
Retention method for system log	By days
Shut down the computer when the security ...	Disabled




Password Policy

Administrator uses those settings for self-discipline. They are not as critical as in human user environment, where users can change their own passwords to obvious one. The most critical settings are enabling complexity requirements and minimum password length (12) to enforce complex administrator and service account passwords.

Policy	Computer Setting
Enforce password history	18 passwords remembered
Maximum password age	60 days
Minimum password age	2 days
Minimum password length	12 characters
Passwords must meet complexi...	Enabled
Store password using reversibl...	Disabled

Account Lockout Policy










To keep up with the main goal of availability, administrator set the Account Lockout Duration to 0. The reference template for this section was hisecweb.inf.

Policy ▲	Computer Setting
 Account lockout duration	0
 Account lockout threshold	5 invalid logon attempts
 Reset account lockout counter after	30 minutes

Audit Policy

AlsAsp audit failures of all event types except directory service (no domain) and process tracking. All file systems have SACL, which enable auditing of failed access to any file - this set through "File System" section of security template. AlsAsp carefully weighted where to enable "Success" audits. Because for local computer authentication successful account logon would always follow by logon event, AlsAsp decided not to log "Success" for account logon events.

"Success" is logged for object access events to enable logging of access to critical files, such as IIS metabase.

Policy ▲	Computer Setting
 Audit account logon events	Failure
 Audit account management	Success, Failure
 Audit directory service access	Not defined
 Audit logon events	Success, Failure
 Audit object access	Success, Failure
 Audit policy change	Success, Failure
 Audit privilege use	Failure
 Audit process tracking	No auditing
 Audit system events	Success, Failure

Few settings in Security Options section were relevant here too.

The "Audit the access of global system objects" option is disabled. If it was enabled it would produce huge and unmanageable audit trail, while placing additional overhead on the system - each mutex, semaphore etc are created with SACL.

The "Audit the use of Backup and Restore privilege" option is enabled to keep audit trail of those important events.

User rights assignment and restricted groups

AlsAsp paid special attention to user rights assignment and group membership. The template was created to assign as limited rights as possible. During

configuration and testing, we found a useful but undocumented feature. When administrator configures user rights and refers group or account that exists on computer, where editor runs, the template editor saves into template file not group or account name, but SID. As soon as it “well-known SID” (such as S-1-5-32-544 for administrators group) the application of template will work on any target computer. However, if this is a custom group the local SID (such as, for example, S-1-5-21-1004336348-1715567821-682003330-1126) will not be applicable on other computer as account with the same name will be assigned there different SID.

However, if group or account does not exist on editor computer, then editor will save name of account into template and will make reusable template, which refers non-standard groups/accounts.

This feature works for user rights assignments and restricted groups sections of template. Unfortunately, it does not work for the sections that involve ACL editor (file, registry, service security). It is mandatory to use well-known principals in ACL editor to allow template to be transferable between computers.

Most of user rights are either limited to administrators group or to no accounts at all. No accounts and groups (except LocalSystem, which implicitly holds those rights) are assigned the following rights: Act as part of the operating system, Add workstations to domain, Create a token object, Create permanent shared objects, Debug programs, Enable computer and user accounts to be trusted for delegation, Generate security audits, Lock pages in memory, Remove computer from docking station, Replace a process level token and Synchronize directory service data.

Only administrators have the following rights: Back up files and directories, Change the system time, Create a pagefile, Force shutdown from a remote system, Load and unload device drivers, Manage auditing and security log, Modify firmware environment variables, Profile single process, Profile system performance, Restore files and directories, Shut down the system, Take ownership of files or other objects.

Using user rights AlsAsp had shaped very specific requirements to logon.

Specifically for this purpose two additional groups were created -

ServiceAccounts, which includes service accounts for SQL Server and application server, and OtherAccounts, which includes accounts that should be denied all logon rights (actually only disabled Guest account and decoy Administrator account).

Groups Web Applications and Web Anonymous Users created by IIS lockdown tool were also used for user rights assignment.

The “Log on as a batch job” right is needed for Web Applications to run isolated IIS process and for Administrators to run privileged scheduled jobs – application of template by secedit.exe, patch level checking by hfnetchk etc.

ServiceAccounts and OtherAccounts are listed in “Deny log on as a batch job”.

“Log on locally” right is required for Administrators to administer server. It is also necessary for Web Anonymous Users if password managed by IIS.

ServiceAccounts and OtherAccounts are listed in “Deny log on locally”.

Only ServiceAccounts granted right “Log on as service”. OtherAccounts and Administrators are listed in “Deny Log on as Service” – mostly as reminder not to set custom service to run as Administrator accidentally even for temporary testing purposes.

Only Web Anonymous Users need “Access this computer from the network” right and they need it only if password for IUSR is managed by administrator.

The particular challenge was presented by “Bypass traverse checking” right.

Low privileged accounts (Web Anonymous Users and Service Accounts) were granted very specific rights on NTFS file system.

In particular, they were not granted any rights on drive roots and had access only to specific directories. Without “Bypass traverse checking” right they could not access those subfolders and generated failure audit records accessing drive roots. While initially this right was granted, it was noticed later that this right allows those accounts to enumerate directory tree without explicit permission to access it. In maximum security environment it might be more secure to grant specific users right to “List folder content” with “This folder only” limitation for specific folders starting from drive root.

In AlsAsp environment this solution was not applicable, because it was impossible to use template to grant such a specific NTFS ACLs for not “well-known” user groups – see above for templates ACL editor limitations.

AlsAsp decided to grant the right “Bypass traverse checking” to low privileged users. The SQL Server service account has demanded option “Increase scheduling priority” right for normal functionality.

Below is a summary of AlsAsp assigned user rights.

© SANS Institute 2003,



Policy ▲	Computer Setting
Access this computer from the network	Web Anonymous Users,Administrators
Act as part of the operating system	
Add workstations to domain	
Back up files and directories	Administrators
Bypass traverse checking	Web Anonymous Users,ServiceAccounts
Change the system time	Administrators
Create a pagefile	Administrators
Create a token object	
Create permanent shared objects	
Debug programs	
Deny access to this computer from the network	OtherAccounts,ServiceAccounts
Deny logon as a batch job	OtherAccounts,ServiceAccounts
Deny logon as a service	Administrators,OtherAccounts
Deny logon locally	OtherAccounts
Enable computer and user accounts to be tru...	
Force shutdown from a remote system	Administrators
Generate security audits	
Increase quotas	Administrators
Increase scheduling priority	ServiceAccounts,Administrators
Load and unload device drivers	Administrators
Lock pages in memory	
Log on as a batch job	Administrators,Web Applications
Log on as a service	ServiceAccounts
Log on locally	Administrators
Manage auditing and security log	Administrators
Modify firmware environment values	Administrators
Profile single process	Administrators
Profile system performance	Administrators
Remove computer from docking station	
Replace a process level token	
Restore files and directories	Administrators
Shut down the system	Administrators
Synchronize directory service data	
Take ownership of files or other objects	Administrators

Below are restricted groups – this is one of few sections which need adjustment between computers, as it refers IUSR and IWAM accounts specific for each server.

Group Name ▲	Members	Member Of
Administrators	AlsAdmin1999	
Backup Operators		
OtherAccounts	NoGuestsAllowed,Administrator	
Power Users		
Replicator		
ServiceAccounts	svcuser3,svcuser2,svcuser1	
Web Anonymous Users	IUSR_<server name>	
Web Applications	IWAM_<server name>	

Security options

Local Policies/Security options had demanded a lot attention. While most of the settings here are obvious for bastion host configuration, there were few settings where AlsAsp has decided against "mainstream".

Policy ▲	Computer Setting
 Additional restrictions for anonymous connections	No access without explicit anonymous permission
 Allow server operators to schedule tasks (domain controllers only)	Disabled
 Allow system to be shut down without having to log on	Disabled
 Allowed to eject removable NTFS media	Administrators
 Amount of idle time required before disconnecting session	60 minutes
 Audit the access of global system objects	Disabled
 Audit use of Backup and Restore privilege	Enabled
 Automatically log off users when logon time expires	Enabled
 Automatically log off users when logon time expires (local)	Enabled
 Clear virtual memory pagefile when system shuts down	Disabled
 Disable CTRL+ALT+DEL requirement for logon	Disabled
 Do not display last user name in logon screen	Enabled
 LAN Manager Authentication Level	Send NTLMv2 response only\refuse LM & NTLM
 Message text for users attempting to log on	This is a Private System. All unauthorized acces..
 Message title for users attempting to log on	Private System Logon
 Number of previous logons to cache (in case domain controller is ...	0 logons
 Prevent system maintenance of computer account password	Disabled
 Prevent users from installing printer drivers	Enabled
 Prompt user to change password before expiration	1 days
 Recovery Console: Allow automatic administrative logon	Disabled
 Recovery Console: Allow floppy copy and access to all drives an...	Disabled
 Rename administrator account	AlsAdmin1999
 Rename guest account	NoGuestsAllowed
 Restrict CD-ROM access to locally logged-on user only	Enabled
 Restrict floppy access to locally logged-on user only	Enabled
 Send unencrypted password to connect to third-party SMB servers	Disabled
 Shut down system immediately if unable to log security audits	Disabled
 Smart card removal behavior	Not defined
 Strengthen default permissions of global system objects (e.g. Sy...	Enabled
 Unsigned driver installation behavior	Do not allow installation
 Unsigned non-driver installation behavior	Warn but allow installation

While most of templates (7 of 9) enable setting "Clear virtual memory pagefile when system shuts down", AlsAsp has set it off, because of concern of longer reboot. Company security risks put the risk of prolonged downtime very high. To use not cleaned pagefile the attacker needs to get physical access to the server and reboot it into other operating system. As this would already constitute the failure of server security, not cleaned pagefile would not add any more trouble. As AlsAsp does not use SMB communications (server service not running) and does not communicate with domain controller, the SMB signing and secure channel options were disabled (this settings also not shown on the picture). AlsAsp has defined rename of Administrator and Guest Account, making them more difficult to guess for potential attackers.

Custom Registry Settings

It is possible to put additional registry settings into template, which are not shown in the template editor. AlsAsp put several such settings into template.

The line "machine\system\currentcontrolset\services\ipsec\nodefaultexempt=4,1" in [Registry Values] set increased security for IPSEC. IPSEC layer does not exempt RSVP and Kerberos from filtering with this setting. It still however exempts broadcast, multicast and IKE traffic. On .Net server (and XP) it will be possible to raise this value to 3 to exempt only IKE. It is also possible that value 3 for this setting will appear in Windows 2000 SP4.

There are also settings to harden IP stack, IIS and NTFS taken from "Securing IIS" SANS institute course by Jason Fossen and Microsoft article "Security Considerations for Network Attacks"

machine\System\CurrentControlSet\Services\Tcpip\Parameters\SynAttackProtect=4,2
machine\System\CurrentControlSet\Services\Tcpip\Parameters\TcpMaxHalfOpen=4,100
machine\System\CurrentControlSet\Services\Tcpip\Parameters\TcpMaxHalfOpenRetried=4,80
machine\System\CurrentControlSet\Services\Tcpip\Parameters\EnableICMPRedirects=4,0
machine\System\CurrentControlSet\Services\Tcpip\Parameters\DisableIPSourceRouting=4,2
machine\System\CurrentControlSet\Services\Tcpip\Parameters\PerformRouterDiscovery=4,0
machine\System\CurrentControlSet\Services\Tcpip\Parameters\EnablePMTUDiscovery=4,1
machine\System\CurrentControlSet\Services\Tcpip\Parameters\NoNameReleaseOnDemand=4,1
machine\System\CurrentControlSet\Services\Tcpip\Parameters\KeepAliveTime=4,300000
machine\System\CurrentControlSet\Services\Tcpip\Parameters\DisableDynamicUpdate=4,1
machine\System\CurrentControlSet\Services\Tcpip\Parameters\EnableDeadGWDetect=4,0
machine\System\CurrentControlSet\Services\W3SVC\Parameters\SSIEnableCmdDirective=4,0
machine\System\CurrentControlSet\Services\W3SVC\Parameters\AllowSpecialCharsInShell=4,0
machine\System\CurrentControlSet\Control\FileSystem\NtfsDisable8dot3NameCreation=4,1

Event Log

Policy	Computer Setting
Maximum application log size	19968 kilobytes
Maximum security log size	19968 kilobytes
Maximum system log size	19968 kilobytes
Restrict guest access to application log	Enabled
Restrict guest access to security log	Enabled
Restrict guest access to system log	Enabled
Retain application log	15 days
Retain security log	15 days
Retain system log	15 days
Retention method for application log	By days
Retention method for security log	By days
Retention method for system log	By days
Shut down the computer when the security audit log is full	Disabled

System Services

AlsAsp found that templates allow a way to configure most of settings relevant to system services. While it is possible to run WMI script to configure most of the

service parameters, the WMI is too powerful a feature to have it running on bastion host.

AlsAsp follows strategy to run as few services as possible and disable all unused services. Only the following base and custom services are enabled on the server to start automatically:

Service Name	Network Usage	Startup Account	Comments
Event Log		LocalSystem	
IPSEC Policy Agent	UDP 500 IKE	LocalSystem	
NTLM Security Support Provider		LocalSystem	
Protected Storage		LocalSystem	
Remote Procedure Call	TCP 135,445, UDP 445	LocalSystem	
IIS Admin Service	TCP 1025	LocalSystem	
WWW publishing service	TCP 80, 443	LocalSystem	
Windows Time		LocalSystem	
Task Scheduler	TCP 1027	LocalSystem	
MSSQL Server	TCP 1433, UDP 1434	Svcuser1	
PCAnywhere	TCP 5631 UDP 5632	LocalSystem	
Plug and play		LocalSystem	
SMTP	TCP 25	LocalSystem	
Security Accounts		LocalSystem	
SQLServerAgent		SvcUser1	
AlsAsp host service	Multiple UDP ports	SvcUser2	

It is important for Windows Time service configuration to ensure that LocalNTP registry setting has its default value 0, which make this service to run as client only.

Task Scheduler service should run as LocalSystem account – if configured otherwise the explicit message demanding LocalSystem account is written into event log and service refuses to start. While Task Scheduler is required for AlsAsp backend operations, it is still very vulnerable service. One of common avenues of attack on this service is its backward compatibility support through at.exe command line tool. While at.exe is secured using NTFS ACLs, there is an additional protection step to be done –set "AT Service Account" to existing account, but wrong (dummy) password - front-end validation validates only

account name. In the Scheduled Tasks window, open the "Advanced" menu and then choose AT Service Account. This will make any jobs scheduled through at.exe interface to fail, while keeping jobs scheduled with Task Scheduler run normally. There is no other way to accomplish this switch, but through the visual interface. The problem was brought to mailing lists and has no solution by now. While Plug and Play service not listed in most of the sources as critically required services, we have found that without this service the system hung on shutdown. As this directly affects availability of system, the service was configured to start automatically.

Several services might be needed on occasional basis. Windows Installer and Server services are needed during initial installation and to install patches. Hfnetchk.exe used for patch management requires Workstation service to run. Those services are configured as manual and administrator can start or stop them when needed. For example, Workstation service is start/stopped from the same batch file running hfnetchk. Administrator manually starts Windows Installer and Server during installation of service packs.

All services not mentioned above are disabled on AlsAsp servers.

While selecting set of services to start is critically important, it is as important as to configure services security. The selected security settings are:

- Administrator – full control
- System – Read/Start/Stop/Pause
- Everyone – Read
- Everyone – Audit Failure

At this point AlsAsp again used SQL server representation of templates. We have configured rights described above with one service and found internal template representation of those rights – opened template file in notepad. Then we generated distinct table of all services mentioned in loaded templates and issued SQL statement:

```
Select servicename + ',4,"D:AR(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO  
;;;BA)(A;;CCLCSWLOCRRRC;;;WD)(A;;CCLCSWRPWPDTLOCRRC;  
;;SY)S:AR(AU;FA;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;WD)' from  
servicelisttable order by servicename
```

Resulting text was pasted into [Service General Setting] section of template.

We got template, which configure all possible services and disable them.

Switching few services, we are interested in, into Automatic and Manual startup was trivial.

Resulting template does not configure such settings as startup account and service recovery options. We also found no command line tools, which would allow this functionality.

AlsAsp has found that this functionality accessible through windows API and written its own command line tool to configure service recovery option. Appendix 1 includes reference to this tool executable and full source code. AlsAsp has standard recovery options for all enabled services, which include running script file to send alert email to administrator.

According to <http://support.microsoft.com/default.aspx?scid=kb;en-us;247929> to execute script or batch file from service recovery options, it should be specified with calling program as "forcedos.exe myfile.bat" or "wscript.exe myscript.vbs". While the process described above produced very explicit and effective template for services configuration, we would be much more comfortable if Security Template Editor would incorporate this functionality. This can be done by ability to mark service configuration record as "default", so it will apply to all not explicitly configured services. The ability to configure service recovery parameters through security template will be helpful too.

Registry and File System hardening

Using SQL Server template analysis it was easy to establish that 'baseline', 'hiseaws' and 'FMStocksP' offer same set of registry security settings. 'cc_server' and 'cc_server_sec' templates are also the same. Registry security settings were selected from 'cc_server', 'baseline', 'basicsv' and 'w2k_server' template. NSA template 'w2k_server' turned out to be the most restrictive and served as basis for AlsAsp template. However, it was missing restrictive settings for few important registry keys and they were copied from 'baseline' template:

MACHINE\SOFTWARE\Microsoft\Command Processor"
MACHINE\SOFTWARE\Microsoft\Ole"
MACHINE\SOFTWARE\Microsoft\Rpc"
MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AEDebug
MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options
MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Svchost
MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip

The most exhaustive list of secured files is in 'basicsv' template, but security ACLs applied to them is not restrictive. AlsAsp choose as basis the set of file system permissions from 'baseline' template and added several more items – most noticeable are items describing security on roots of not system drive. This security is set in mode "Propagate inheritable permissions to folders and files" and allows full access for administrator and system, while enabling failure audit for everyone. It is safe to reapply template again without risk to disrupt explicit permissions for subfolders.

Template also restricts permissions on MetaBack folder to protect IIS metabase backups.

For full list of registry keys and files secured see AlsAsp template provided in Appendix 1.

IIS Security

IIS is a major component of AlsAsp security architecture and securing IIS is a critical task. The following IIS specific actions were taken:

- Application of latest service pack and security patches for IIS got highest promotion priority, as IIS is the most exposed service. The usual escalation time for those patches is minimal (see Patch Application Process for details)
- All volumes are formatted with NTFS and have default ACLs applied at root allowing access only to true (not decoy) administrator and system account
- All IIS requests are logged and all logs are archived
- Content files are installed on separate NTFS volume E:
- All sample/help files deleted and correspondent directories secured by ACL to prevent reinstall.
- Dangerous executables are protected. Unfortunately, the recommendation quoted in multiple sources "delete or move to another location" does not work for these files, because Windows File Protection Service will restore them almost immediately. While it is possible to disable Windows File Protection (see <http://www.griffin-digital.com/wfp.htm> for example), it is a big blow to server security, stability and virus resistance. That is why the only effective way to secure dangerous executables is to place restrictive ACLs on them. See File Security section for details.
- The option Write unsuccessful client requests to Event Log should be enabled
- IIS lockdown tool was run and URLScan was installed.
- All applications are running in high isolation mode under IWAM account.

IIS Lockdown tool

At the beginning of installation process (before running template) AlsAsp choose IIS Lockdown tool available for download from <http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/tools/locktool.asp> and applied it with template "Dynamic Web Server (Asp Enabled)". The wizard has asked what services need to be disabled and disabled SMTP, FTP and NNTP without giving chance to enable SMTP, which needs to run on the server. Then it asked what unused script maps should be disabled - only ASP was left. On Additional Security page Wizard asked to remove dangerous virtual directories from IIS installation (IIS Samples, MSADC, IISHelp, Scripts, IISAdmin); set file ACLs on system utilities (cmd.exe, tftp.exe - full list not provided); set ACLs on content directories to prevent anonymous IIS users to write to them; and disabled WebDAV. Last wizard page allowed installing URLScan IIS filter on the server. Oblt-log.log was generated by the tool. This file clearly shows what wizard done:

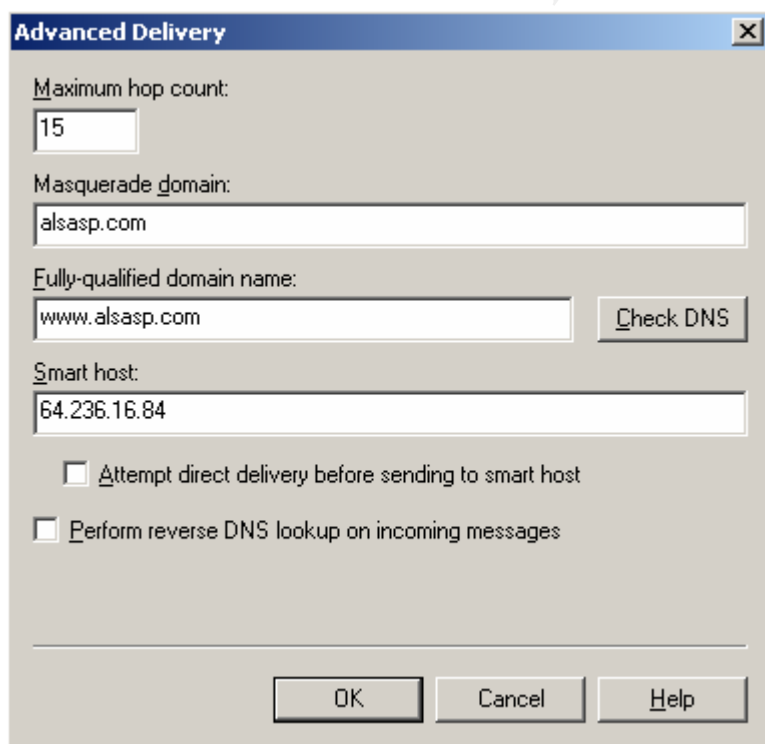
1. Created groups Web Anonymous Users and Web Applications and added IUSR to IWAM account to those groups correspondently (groups named _Web Anonymous Users and _Web Applications if computer is on domain).
2. Reset service startup settings
3. Backed up metabase
4. Installed URLScan

5. Remapped ISAPI extensions to 404.dll
6. Placed secure ACL on unused ISAPI DLLs (httpext.dll, idq.dll)
7. Denied access to Web Anonymous Users and Web Applications to all EXE and COM files in c:\winnt folder, except dllhost.exe access for Web Applications – this file is needed to run pooled/isolated application protection mode.
8. Denied write access to c:\inetput\wwwroot folder and all content folders for all virtual directories to Web Anonymous Users and Web Applications groups.

Some of those setting need regular refresh to ensure that they are still here and they are included them into regularly applied hardening script and template. Out of the box, URLScan came configured with values satisfactory for AlsAsp. We changed the only setting: AlternateServerName=ABCSEVER. This change makes IIS to report its version in HTTP headers as bogus string and defeat simple scripts scanning for IIS server to attack.

SMTP Security

While IIS lockdown wizard disabled SMTP service AlsAsp re-enabled it, as business functionality include requirement to send email messages from application. The main security problem with SMTP service is that by default it requires two-way communications with all possible destinations – over TCP port 25 (SMTP). The firewall rule controls that TCP session should be initiated from internal interface, but it is just one layer of protection. Should firewall logic fails, any packet with source of port 25 can reach the server.



AlsAsp has increased security of SMTP service by using mail server of hosting ISP. Hosting ISP mail server (not managed by AlsAsp and serving multiple clients as their email server) on AlsAsp request was configured to allow relay for AlsAsp subnet. AlsAsp SMTP servers in turn were configured to use "Smart Host" option to relay all outgoing mail to mail.my_isp.com (actually the smart host option was configured to IP address to save on DNS lookups - assuming that mail.my_isp.com permanently configured with IP address).

In this way, the only connection to port 25, which should be opened on firewall and IPSEC, is a connection to trusted and secured ISP email server. It does not mean that this server cannot be compromised and used to stage attack against AlsAsp, but risk is transferred and significantly reduced.

AlsAsp completely logs all SMTP activity and backup those logs.

PCANYWHERE – Securing Remote access

Historically AlsAsp used Symantec PCAnywhere for remote administration. When company created its security architecture, it reviewed this choice of remote administration tool and decided that it matches company's security goals.

PCAnywhere is highly functional tool, which allows for complete remote control of desktop, highly optimized transfer of remote desktop image over the network, file transfer, reboot of remote server. PCAnywhere has produced by one of the most respected security software companies – Symantec.

PCAnywhere is rarely mentioned as secure remote administration tool. Choices that are much more traditional are Terminal Services, remote share access, VNC and VPN. PCAnywhere provides complete control over host desktop and optimized file transfer capabilities. While pricey it is not cost prohibitive for small-scale installations especially using discounts for not recent version (AlsAsp is using 10.0 while the latest is 10.5). The excellent article by Mark Burnett "Remote Management of Win2K Servers: Three Secure Solutions"

(<http://online.securityfocus.com/infocus/1629>) lists those solutions as well as important criteria required for remote administration tools - access control, integrity, confidentiality and auditing. We demonstrate that those criteria do apply to PCAnywhere in described configuration. There are also multiple redundant layers for each criteria, which comply with defense in depth strategy.

Access Control/Authentication

AlsAsp implements access control for remote administrator access using the following layers.

1. Firewall authentication – firewall policy require authentication to allow connection on PCAnywhere port.
2. IP Address filtering on firewall, IPSEC and PCAnywhere configuration allows connection only from predefined IP addresses.
3. Host serialization - latest feature in PCAnywhere (started from v10.0) provides host level authentication using what Symantec named "serialization". This is a shared secret assigned to installation of PCAnywhere. The same secret shared by hosts and remote installations.

Only installations with the shared secret being the same can connect to each other.

4. While it was possible to reuse user accounts for PCAnywhere authentication it was decided to add one more layer of authentication with login/password different from windows login/password and different for each host. This creates additional step for attacker to overcome – before getting to windows logon prompt the correct PCAnywhere credentials need to be entered. To prevent password guessing host service configured with "Limit number of login attempts" option set to 3.
5. Host service configured with option "Logout windows on disconnect". This unavoidably requires user to enter windows login/password into Windows login prompt as well as be subject to all windows account lockout and audit policies.

Confidentiality

AlsAsp has configured PCAnywhere with symmetric encryption level on each side and option "deny lower levels" checked. Several support articles (<http://service1.symantec.com/SUPPORT/pca.nsf/pfdocs/1999022312571812>) and answers by Symantec customer service have clarified some internals of encryption. Immediately before each connection the host and remote generate separate pair of public and private keys and exchange public keys - the host sends its public key to the remote and the remote sends its public key to the host. The host encrypts its data stream with the remote's public key and the remote encrypts its data stream with the host's public key - correspondent private keys used for decryption. Using this agreement host generate 128 bit symmetric key, which then used for bulk encrypting of one and only one session. The logon prompt and credentials transfer happen over already encrypted channel. While key exchange procedure is susceptible to men in the middle attack, this is mitigated by authentication steps (see above).

PCAnywhere also supports certificate authentication level. The only difference with symmetric is that instead of initial public key exchange hosts use trusted certificates. However, there were several reports about instability of this option and AlsAsp decided to trade off additional security for more stable option.

Integrity

Unfortunately, Symantec declined to provide information about packet integrity features in PCAnywhere. However, look on code dependencies shows calls from PCAnywhere to CryptSign function of CryptAPI, which suggest that either SHA1 or MD5 hash does appended to each encrypted data packet. PCAnywhere also has its own "tripwire" like feature named "Integrity checking". This feature verifies consistency of binary files and configuration settings.

Auditing

PCAnywhere has very detailed audit functionality. It allows write audit information about every file and program that is accessed during a remote control session for security and auditing purposes directly into windows event log.

Identified security problems

Unfortunately, it is impossible to configure PCAnywhere to run under account with privileges lower than LocalSystem. The main reason behind this is a requirement to access Window Station kernel object not available to anyone except interactive user and LocalSystem (this is the same reason why "Allow service to interact with desktop" checkbox in Service administration snap-in available only for LocalSystem account).

PCAnywhere has excellent vulnerability history. Over last years PCAnywhere had only 4 minor security problems, all fixed in previous versions. The high reputation of Symantec in security business is also adding credibility to the quality of this solution.

MSSQL Security

Issues and Goals

While SQL Server never exposed for direct access of outside users and has at least 2 layers of defense above it (firewall and IPSEC) the security of SQL Server is a crucial piece of the whole security architecture. As said above the specific of AlsAsp is that SQL Server is running on bastion Internet faced server and possibly on the same server as IIS.

When protecting SQL Server the following attack vectors considered:

- Unauthorized access to connection endpoints with intent to explore buffer overflow or DoS attack
- Unauthorized access to data by logging in to the server
- Elevation of privilege by allowed valid user connection, such as SQL injection.
- Fraud access to raw data files and backups

The security plan for SQL Server was developed based on author DBA experience, Microsoft OpenHack documentation, Microsoft SQL 2000 C2 Guide and excellent checklist from www.sqlsecurity.com.

The security of SQL Server is configured assuming just two different access types - sysadmin/administrator/superuser and low-privileged user accessing database from web application or application server. Its policy requirements for all applications to use only stored procedures for database access, no direct table access allowed. Even if potential attacker gains access to database, he would only be able to issue the very same set of calls he can issue from

application, however in possibly different order and with different parameters. With measures taken to prevent access to stored procedures source code it is almost impossible to incur more damage to database or learn more data than while accessing database from within application.

Each application has separate set of credentials and as such different connection string. If attacker penetrates one of application windows accounts he will be at most capable to call stored procedure assigned to correspondent application. He will need to get credentials of different account to gain access to another set of stored procedures.

While it is possible to limit sysadmin account, it should be done very carefully as it might break functionality and will most likely break installation of upgrades and service packs. In addition, in most cases, sysadmin can restore previously limited access. That is why we have assumed that all possible measures should be taken to prevent unauthorized sysadmin access to SQL server and have taken no measures to limit authenticated sysadmin account.

SQL Server Installation configuration

The starting point to secure MSSQL is to install it securely. During installation the minimum required feature set was installed by un-checking replication support, full-text search, debug symbols, development tools (as soon as feature not installed it cannot be explored).

As said above it is important to run software under as low privilege account as possible. SQL Server is giving this possibility, but not from installation - if low privilege user account selected during installation the installation will fail.

There is a requirement to run LM Server and Workstation services during installation and initial configuration of the server, they are not required for production run of MSSQL.

All file systems on IISAs servers are NTFS and as such SQL Server installed on NTFS partition.

During installation, MSSQL and SQLServerAgent are installed under LocalSystem account. The critical step that mitigates almost all others is to apply latest service packs and patches. The service pack 3 (latest at time of writing) applied to installation as well as post SP3 patches (none by the time of writing).

At this point, the account to run MSSQL and SQLServerAgent should be switched to low privileged service account **svcuser1**. This service account is used only by SQL Server and not shared with any other applications. This should be done using Enterprise Manager (EM) while being logged under real administrator account. In this case, EM automatically applies permissive permission to proper folders and registry keys and grants this account the right to logon as service. To harden this account further the "Act as part of the operating system", "Log on locally" and "Log on as batch job" privileges should be removed from this account.

Standard service recovery features applied to MSSQL Service and SQLServerAgent (see service recovery).

During the change of service account, Enterprise Manager tool applies permissions to file system and registry, which allow for normal functionality. However, it is possible to harden them further.

Permission for **svcuser1** on c:\program files\Microsoft SQL Server\MSSQL directory can be changed from Full to Read/Execute, while keeping full permission on c:\program files\Microsoft SQL Server\MSSQL and write only permissions on c:\program files\Microsoft SQL Server\Log (folders provided for default installation). This limitation should stop attacker who gained identity of **svcuser1** by possible buffer overflow in MSSQL from modifying binary files and deleting log files. There is no permission for this account outside SQL directories. Same type of hardening can be done on registry.

While Full permission has been granted to **svcuser1** to all sub-keys of HKLM\Software\Microsoft\MSSQLServer key, it is enough to have read access to all sub-keys except HKLM\Software\Microsoft\MSSQLServer\MSSQLServer.

Connections and user authentication

MSSQL 2000 Server allowed for multiple connection methods by meaning of installing network libraries in server network configuration. AlsAsp supports two configurations.

SQL Server running on the same computer with data consumers (IIS, application server)	SQL Server running on the different computer with data consumers (IIS, application server)
No network library has been installed and IPSEC rule allowing connections on TCP port 1433 is unchecked. The only way to connect to the MSSQL 2000 Server in this case is using shared memory interface from the same computer. This configuration change by itself eliminates the whole class of remote exploits.	The choice of network protocol must be made. The usability and performance criteria leave Named Pipes and TCPIP libraries (others including SSL encryption are much slower). To use named pipes however the LM Server service should be running and dangerous port 445 has to be made accessible. This argument is leaving TCPIP access as front-runner. MSSQL 2000 Server will listen on TCP port 1433 and UDP port 1434 on all available interfaces. There is no way to limit it listening to only intranet interface. To minimize exposure those ports are unconditionally closed on firewall and secured by IPSEC (see firewall and IPSEC sections).
In both cases, SQL Server was configured in windows authentication mode. This means that all standard OS account/password management features applied, such as password complexity, account lockout etc. This option is the only allowed for SQL Server C2 certification and is as secure as Windows 2000. See note below.	
This account does not need	As AlsAsp run standalone computers, the

access from network rights. The IUSR account is granted access to SQL Server.	integrated authentication will use NTLM challenge/response dialog to verify credentials. Local user accounts with the same name and password should be used on both computers. For IIS this means that web server should be configured to use custom user account instead of standard IUSR (it was chosen not to rename IUSR account to avoid metabase incompatibilities). Computer roles in AlsAsp are interchangeable. This account should be granted access this computer from network for SQL Server access and logon locally rights for IIS to be able to call LogonUser.
Credentials of account already authenticated during caller startup and no overhead required authenticating further.	This authentication is slower. However, effective connection pooling almost eliminates this problem.

Note about back-runner - SQL Mixed (Native) authentication

To verify claims about insecurity of native SQL Server authentication we run few experiments. The passwords for the SQL Authentication are sent over the network using a very weak password encryption method. This was first mentioned in David Litchfield's paper "Threat Profiling Microsoft SQL Server" (<http://www.nextgenss.com/papers/tp-SQL2000.pdf>). In his paper, Mr. Litchfield states that the password is encrypted by first converting it into UNICODE and then performing a simple XOR operation.

We did our own Network monitor trace of user connecting with "secure" login "test/pwd" and found this description a bit incomplete. Full process is:

1. Capitalize password string to PWD.
2. Convert string to Unicode (00 70 00 77 00 64)
3. Reverse on each byte 4 most significant bits with 4 less significant (00 07 00 77 00 46)
4. XOR each byte with A5 (A2 A5 D2 A5 E3 A5).

The picture illustrates this:

```

+Frame: Base frame properties
+ETHERNET: ETYPE = 0x0800 : Protocol = IP: DOD Internet Protocol
+IP: ID = 0x3E91; Proto = TCP; Len: 212
+TCP: .AP..., len: 172, seq:3070063098-3070063270, ack: 847001512, win:17483, s

```

```

00000000  00 B0 D0 B6 EE FD 00 E0 B1 47 36 67 08 00 45 00  .|||e².αG6g. E.
00000010  00 D4 3E 91 40 00 7F 06 A1 58 0A CA 03 13 0A 17  .>ae0.ó±iXollv!!0t
00000020  03 47 0E 57 05 99 B6 FD 71 FA 32 7C 37 A8 50 18  ▼C#0±*π^S;úúP†
00000030  44 4B 8C 26 00 00 10 01 00 AC 00 00 01 00 A4 00  DKi&...@.W...@.ñ.
00000040  00 00 01 00 00 71 00 00 00 00 00 00 00 07 BC 19  ..@..q.....|+Login
00000050  00 00 00 00 00 00 E0 03 00 00 68 01 00 00 09 04  .....αv...h...o+Password
00000060  00 00 56 00 00 00 56 00 04 00 5E 00 03 00 64 00  ..V...V.+.^+v.d.
00000070  12 00 88 00 0A 00 00 00 00 00 9C 00 04 00 A4 00  t.ê.@.....ê.+ñ.
00000080  00 00 A4 00 00 00 00 06 5B 1B B0 FC 00 00 00 00  ..ñ....+[-+^.....
00000090  A4 00 00 00 75 00 73 00 65 00 72 00 A2 A5 D2 A5  ñ...u.s.e.r.óñτñ
000000A0  E3 A5 53 00 51 00 4C 00 20 00 51 00 75 00 65 00  nñS.Q.L..Q.u.e.
000000B0  72 00 79 00 20 00 41 00 6E 00 61 00 6C 00 79 00  r.y..A.n.a.l.y.
000000C0  7A 00 65 00 72 00 31 00 30 00 2E 00 32 00 33 00  z.e.r.l.O...2.3.
000000D0  2E 00 33 00 2E 00 37 00 31 00 4F 00 44 00 42 00  ..3...7.l.O.D.B.
000000E0  43 00                                     C.

```

For comparison below is intercept of integrated SQL login authentication exchange:

```

00000000  00 B0 D0 B6 EE FD 00 E0 B1 47 36 67 08 00 45 00  .|||e².αG6g. E.
00000010  00 D6 61 B1 40 00 7F 06 7E 36 0A CA 03 13 0A 17  .>ae0.ó±~6ollv!!0t
00000020  03 47 0E EF 05 99 F1 A7 CB FC 53 3B 96 A3 50 18  ▼C#0±*π^S;úúP†
00000030  43 90 23 FE 00 00 11 01 00 AE 00 00 01 00 4E 54  CÊ#...@.<...@.NT
00000040  4C 4D 53 53 50 00 03 00 00 00 18 00 18 00 76 00  LMSSP.v...t+.v.
00000050  00 00 18 00 18 00 8E 00 00 00 16 00 16 00 40 00  ..t+.Ä...-.-@.
00000060  00 00 0A 00 0A 00 56 00 00 00 16 00 16 00 60 00  ..@.@.V...-.-.
00000070  00 00 00 00 00 00 A6 00 00 00 05 82 80 A0 55 00  .....²...áÇáU.
00000080  53 00 57 00 4B 00 44 00 51 00 4B 00 53 00 50 00  S.W.K.D.Q.K.S.P.
00000090  30 00 31 00 74 00 65 00 73 00 74 00 31 00 55 00  0.l.t.e.s.t.l.U.
000000A0  53 00 57 00 4B 00 44 00 51 00 4B 00 53 00 50 00  S.W.K.D.Q.K.S.P.
000000B0  30 00 31 00 A1 9D 40 B9 84 6E 05 C0 5B 42 CE 9D  0.l.i.v@|án&L[B|v
000000C0  5F DC 21 2C 91 66 4E DA E9 BC 0A 60 91 46 25 C2  _!,&fN-@|@'æf*T
000000D0  68 31 13 E9 4E 42 88 DD 90 A6 0D 96 38 0D 14 67  hl!@NB&| É²&ú8&¶g
000000E0  4C 5D B0 5A                                     L|Z

```

In addition, we tested one of penetration testing tools - forceSQL.exe downloaded from <http://www.nii.co.in>. With SQL authentication, it is quickly breaking passwords while guessing up to 40 passwords per second. While SQL Server is running in Windows authentication mode it still has a login configured, while inactive. The safe practice is to set on this account very complex (72 characters) password using sp_password command – included into hardening script.

Notification of administrators and auditing

SQL Server is a very feature rich system, which has 2 default integrations with messaging as well as new SQL Notification Services. It is also possible to enable “C2 auditing mode” (created for compliance with C2 trusted configuration) when SQL Server logs ALL events.

However bastion host has no mail client installed and does not require full C2 compliance. To satisfy notification and auditing needs the SQL Agent was

created. This job is running Com object CDO.Message that generates email message to administrator. While running under **svcuser1** account this functionality does not require access to scripting executables (wscript.exe, cscript.exe and wsh.exe) as SQL server call Com object directly. The notification is sent on critical data processing problem, intrusion events etc. The SQL Server configured to log all failed access to NT security log (while it is also possible to log all successful connections, it is unfeasible with web access). An alert is configured on SQL Server, which run on failed login events, launch notification job and notifies administrator.

Database permissions

After account is authenticated by MSSQL 2000 Server, it applies complex set of internal security to check permission and authorize data access and actions. The following logins allowed access to SQL Server – renamed administrator, **svcuser1** - account under which SQL Server itself running, **svcuser2** for application server access and **IUSR** account (or other web anonymous account) for the access from Active Server Pages.

It is important to delete BuiltIn\Administrators group from allowed logins, because LocalSystem account is implicit member of this group. While very likely result of buffer overflow is gaining LocalSystem permissions, it is an advantage not to allow LocalSystem account to logon to SQL Server.

To specify security rights SQL Server maps accounts to database users, database roles and server roles.

Administrator and svcuser1 mapped to server role "system administrator" and as such have full and unrestricted access to all server data and functionality.

Access to AlsData database

The rights of svcuser2 and IUSR are very limited. The first step in configuring user rights is to define what databases user will have access. svcuser2 and IUSR have access only to AlsData database. While it is possible to fine-tune SQL security by specifying exact access rights per object it is much more convenient to use database roles.

In this database they are assigned to 3 roles – public, db_denydatareader and db_denydatawriter. Those roles effectively mean that user will not be able to access any data table directly. The advantage is that this denial automatically applies to each new table.

Unfortunately, there is no role in SQL Server, which actually corresponds to Microsoft recommendations –we would name it db_execute – and which will be by default allowed to execute all stored procedures in SQL Server.

To compensate for this SQL Server allowed for creation of custom roles. AlsAsp is using a script (Appendix 1), which creates the custom role and grants execute permission on all stored procedures in database to this role. Then svcuser2 and IUSR are granted access to this role. The disadvantage of custom db_execute

role comparing with standard role is that as soon as new stored procedure created the access should be granted explicitly and failure to do this will result in error equivalent to DoS. That is why we contacted Microsoft several times requesting inclusion of this feature to future releases of MSSQL.

The next step to limit right or svcuser2 and IUSR on AlsData database is to deny then “statement permission” which is a right to create procedures, functions etc. The statement to this is “deny all to db_execute”.

The next step is to deny to db_execute all rights on system tables in AlsData database by issuing commands like “deny all on sysobjects to db_execute”. This will disallow, for example, direct access to stored procedures source code through system tables.

Access to other server databases

While explicitly allowed access to only one database user actually has access to all other databases on the server under special identity guest. To eliminate this possibility the hardening script has removed guest user from almost all other databases – the command is sp_dropuser “guest”. One of side effects of this change is excluding possibility of user to access special msdb database, responsible for scheduling tasks. Users will not be able to schedule or stop running tasks.

It is impossible however to remove guest access from 2 crucial database – master and tempdb. Master database is a huge repository of powerful stored procedures. To deny access to this functionality the hardening script revokes all permission for a guest user and public user making master database inaccessible.

Important exception here is to allow then read access to one table (spt_values) and execute access to one stored procedure (dbo.sp_MShasdbaccess) as they are required for logging into server.

Results of SQL Server hardening

The changes made above completely limit all known privilege elevation attacks, possibility of source code access (to stored procedures), as well as possibility of large-scale unauthorized data changes. See also web application security for additional checks and controls inside SQL Stored Procedures.

Application Security

Web application security

Without touching all aspects of web application security, the following are the most important ones. AlsAsp web application is Active Service Pages application with significant client side scripting code.

All user input is not trusted. Verification of user input happens on several layers. Inside browser's code the input undergoing minimum verification for logical consistency. In URLScan security module the input is verified to be less than reasonable size. In server side ASP code the client side verification happen again and more sophisticated business rules applied. In most cases the user input parameter is sent to SQL Server as parameter value for stored procedure. On this step it is validated by ADO to satisfy parameter type. Stored procedures in turn validate most of parameters for business logic constraints.

One of required verification steps done in server side ASP code is stripping possible script tags to prevent cross-site scripting. AlsAsp achieves this by using custom function GetRequest("parametername") wrapping stock server side object Request. This function has mandatory verification steps.

AlsAsp web application is using form based password authentication against password stored in database. In fact database store not a password, but MD5 cache of password. This prevents a massive security problem of disclosing all application accounts as MD5 cache not reversible.

The ASP script, which verify password converts it to MD5 cache before submitting as stored procedure parameter to database for verification.

AlsAsp implemented inside web application several password security rules, modeled from native Windows password security.

When user sets or changes a password, the server side logic applies complexity filters to a password and requires minimum length of 8 characters, mix of letters and numbers and difference with user name.

The account lockout rules are in place, function the same way as windows account lockout, and will lock account for 30 minutes after 3 failed login attempts.

The email will be sent to administrator about account lockout.

After secure authentication happen over SSL encrypted connection, the user session for most of users run unencrypted over HTTP. The session cookie is issued in random and stored on user computer and as row in database with other session related information. The risk of session hijacking is lowered by highly randomized session number, but still exists and accepted.

AlsAsp is also offering additional security for its users using certificate authentication – if this mode enabled for user account then user machine must have certificate issued by AlsAsp installed, and session fixation – if enabled then user must connect from allowed list of IP addresses. Last measure can increase protection against session hijacking.

AlsAsp is immune from SQL injection attacks by accessing database only through ADO command object and database stored procedures.

AlsAsp web application has intensive session accounting – user can see record of her session and report to administrator suspicious activity.

As web application is running in isolated account in case of buffer overflow the highest privilege attacker can obtain is highly restricted IWAM account credentials. Many measures described elsewhere in this paper assure that elevation of privilege cannot happen.

Other applications security

Application host service is responsible for accepting incoming UDP messages, translated them and storing into database. The host service is programmed defensively using respected compiler by Borland. All input buffer sizes are verified. However, AlsAsp understand possibility of buffer overflow attacks on this service using incoming UDP message.

The service is running under very low privileged account with read-only access enabled only for few keys in registry with configuration information and actual executable file. The access to SQL Server enabled for this account, but the only privilege granted is to call one stored procedure in one database.

With such limitations, the buffer overflow on this service can cause minimum damage.

Operations Security

Problem recovery and business continuity

Business continuity is the first priority for AlsAsp. It addressed in following manner. AlsAsp has configured all servers identically, but has different services subsets enabled. Two servers WEB1 and DB1 serve production load by splitting application roles as web/listeners and database servers.

If one of those servers fails then functionality can be served by one of these servers using several simple configuration changes – administrator intervention required.

BKP1 server is powered down all the time, except when it needs to be upgraded. AlsAsp has IP managed power device with tunnel opened to it on firewall only from admin workstations. The purpose of this server is to pickup load in case of both WEB1 and DB1 servers are failed.

ALPHA server has multiple roles. It hosts all web/listeners/db servers and provide QA platform for application upgrades. Application patches first deployed on this server and tested by QA. This server is also the first to receive vendor software patches. After each application patch QA team has opportunity to test application functionality.

Third role of ALPHA server is a file server for the farm. However, it does not run server service – see backup procedures below.

While less powerful the ALPHA server can be configured to pickup production load, providing additional level of redundancy.

All those redundancy method require administrator intervention. Growing AlsAsp is reviewing clustering options to provide immediate redundancy for critical services.

Backup procedures

The regular backup of data is a critical piece in operations security. AlsAsp is doing frequent – each 6 hours - database backups to a local folder on database server. Log backup created each 15 minutes. Those frequent backups allow recovering of database up to any point in time in case of accidental data corruption. Those backups do not allow recovering database in case of unrecoverable file system failure as they kept on the same RAID array with database.

Each 24 hours the copy of database backup is moved out from main server to ALPHA server. The database backup is a primary to move, but not only file backed up from the server. Also included into backup are:

- System state backup created using Windows Backup Accessory
- Yesterday web server log
- Yesterday SMTP log
- Dumps of Windows Event logs created using resource kit dumpel.exe

The scheduled task on main server generates those files and compresses them using pkzipc.exe into one file with name derived from current date. The destination file is accessible through IIS. IIS expose this directory using additional website running on port 88. Firewall, IPSEC, and IIS list of allowed addresses limit visibility of this web site to AlsAsp subnet only.

Another scheduled task run on ALPHA server an hour later (10 times more than enough for first task to complete – tasks not otherwise synchronized). This task runs wget.exe to download compressed daily backup file from main server to ALHPA server. Success of backup process together with resulting file size is reported in email to administrator.

These procedures allow not to run potentially exposed Server service for unattended file transfer.

The folder that accepts backup files is in turn backed up to tape – AlsAsp has one tape device attached to ALPHA server.

Administrator downloads the latest daily backup file to administrator workstation once a week, where it burned to CDROM. This download is manually initiated, as it was decided to be advantageous to have administrator regularly visit server and personally verify state of backend processes. AlsAsp is reviewing it now to allow scheduled process from administrative workstation to download backup file using wget.exe and burn CD automatically. CDROM backups have indefinite retention.

Such multiple layer procedures allow for very fast recovery of corrupted data from the most recent backup and ability to recover database state and critical files up to any point in company history (with respective precision – for old files the precision is one week).

Patch management

AlsAsp understands that one of the most critical steps in windows server security is to keep servers properly patched and established following patch management

process. The administrator subscribed to security mailing lists from Microsoft, SecurityFocus (bugtraq, ms-focus), SANS. Those lists deliver up to minute notification about available patches. The patch is evaluated and depending on criticality level applied to ALPHA (QA) server. This can happen from few hours to no more than 3 days after announcement. The delay usually reasoned by need to monitor negative feedback on this specific patch and chance to group not emergency patch with next patch - to minimize server reboots. After patch applied to QA server, the application functionality is evaluated by QA stuff. At this point, the patch is ready to be promoted to 2 production servers. The patch promotion usually causes server reboot and 2-minute downtime. Web application specifically designed to ignore this downtime gracefully. AlsAsp still try to minimize those downtimes and group updates together.

A week or two later, after production servers running stable with new patch (sometimes two or three over this time); the backup server is powered on, updated with all patches and powered off again. This delayed update allows for immediate patch roll back – the functionality still not natively available for Microsoft patches. In case of rollback, the backup server should replace both production servers and allow administrator rebuild them.

The critical piece in patch management is management control. Each 2 days the scheduled task runs hfnetchk.exe on production server. The results emailed to administrator and company management. This makes administrator work visible and controlled and not allows delaying patch application for too long. The same job reapply security template using secedit.exe. This ensures that settings are not accidentally changed and modeled after regular group policy application in domain environment.

The patch management is a time consuming task. AlsAsp is testing Microsoft Automatic Updates service, but not ready yet to trust patch application process to Microsoft. AlsAsp had also looked on some management security providers, such as <http://www.sintelli.com/>. This company doing very nice job to prioritize and track patches tailored to specific environment. The price level on this service however is well above abilities of small company.

Security QA – penetration testing

The author's background in software architecture does not allow security architecture to be treated as completed until not tested. AlsAsp used industry standard (and free) testing using security scanner Nessus (<http://www.nessus.org>). This application runs from Linux workstation. The Nessus used "plugins" specifically written to check for specific vulnerabilities. The Nessus tests for practically all known security vulnerabilities and updated regularly. Normally the only vulnerability Nessus reports for AlsAsp server is low risk vulnerability:

"General: HTTP Server type and version www (80/tcp). The remote web server type is ABCSERVER"

REFERENCES

1. SANS Institute, GIAC courseware
2. This paper downloads <http://tarasul.home.attbi.com/GiacPaperATarasul.zip>
3. Howard, Michael, Designing Secure Web-based Applications for Microsoft Windows 2000, Microsoft Press, 2000.
4. NSA Security Recommendation Guides <http://www.nsa.gov/snac/index.html>
5. The Security Operations Guide for Windows 2000 Server
<http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/prodt ech/windows/windows2000/staysecure/default.asp?frame=true>
6. SQL Server 2000 C2 Administrator's and User's Security Guide
<http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/sql/maintain/security/sqlc2.asp>
7. Building and Configuring More Secure Web Sites
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnnetsec/html/openhack.asp>
<http://www.sqlsecurity.com/DesktopDefault.aspx?tabindex=3&tabid=4>
8. Security Considerations for Network Attacks
<http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/prodt ech/windows/iis/dosrv.asp>
9. Microsoft Common Criteria Windows 2000 Common Criteria Security Target
<http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/issu es/W2kCCSCG/default.asp>
10. Brian McKenney, Defense in Depth
http://www.mitre.org/pubs/edge/february_01/mckenney.htm

Appendix 1

To minimize this document size the author made additional files available for download as single file from his private web site at
<http://tarasul.home.attbi.com/GiacPaperATarasul.zip>

This file includes:

- Security Template for AlsAsp
- Command line utility to configure service recovery option (with source code)
- WinNT.sif used for consistent installation process.
- Script to load template files to SQL Server with sample queries
- SQL Server Script to create the custom role and grant execute permission on all stored procedures in database to this role.

© SANS Institute 2003, Author retains full rights.