Global Information Assurance Certification Paper

Attack Correlations

Practical Assignment Version: 1.4

Option 2

Edward Johns

July 1, 2002

Abstract

This paper describes a process that an actual corporation undertook to demonstrate correlations between threats from potential outside attacks. The definition of the threats and countermeasures was based on a methodology called *Attack Tree*. The attack tree methodology breaks down threats hierarchically into sets of scenarios and allows countermeasures to be defined. The ability to structure threats and countermeasures was the first step towards identifying attack correlations. This paper defines a subset of the attack tree scenarios and uses them to demonstrate attack correlations. A workflow was generated as part of this process to identify and implement these attack correlations. Actual attack correlations are identified and explained in the context of the attack tree scenarios presented.

This paper starts with a brief definition of the initial security baseline of the specified corporation. This baseline is the start for the attack tree analysis and drives the content of the attack tree scenarios. The analysis phase allows the countermeasures to be defined that are utilized in the attack correlation workflow. The outcome of the analysis is the implementation of the countermeasures in general and the attack correlations specifically.

Corporate Security Baseline

This corporation based their security policies and procedures on the bedrock principles of *Confidentiality*, *Integrity* and *Availability*. Confidentiality guarantees that corporate information is not disclosed to any unauthorized person whether inside or outside of the corporation. Passwords should not be disclosed to anyone but the user. Integrity ensures that the information and resources owned by the corporation are not compromised in any manner. Unauthorized stealing or manipulation of a password is strictly forbidden. Availability ensures that the corporation is allowed to deliver their resources and services to their employees without interruption. If someone inadvertently or illegally crashes the domain controller that supplies the password authentication, employees can no longer access the resources and information of the corporation.

Enterprise security depends on guaranteeing the confidentiality, integrity and availability of corporate resources, e.g. e-mail systems, DNS servers and firewalls. This corporation was confronted with a myriad of security concerns and relied on a “defense in depth” approach. Defense in depth is a concept that defines a layered approach to protecting data and resources. The data and resources inside a corporation would be protected by multiple mechanisms.

The corporation in question had implemented a limited "defense-in-depth" approach:

- border router,
  Routers forward traffic between networks and can perform limited traffic filtering. A border router was placed on the Internet and used to prevent any traffic from entering the facility that did not pass its limited checks.
- proxy firewall,
  Firewalls protect resources on a private network by filtering network traffic sometimes state-fully and sometimes through a proxy server. This corporation used a proxy firewall to manage traffic flow to/from the Internet. The proxy firewall was considered safe since it handled all inbound and outbound connects to the Internet.
- stable network,
  The corporate network was based on TCP/IP and was considered stable and well maintained. There were no tools available that monitored the internal network for peculiar traffic behavior.
- user authentication,
  The administrator initially issued users their passwords but there was no corporate-wide password policies in-place.
- corporate anti-virus solution and
  The corporation had a corporate-wide anti-virus policy but had no mechanism in-place to correlate virus attacks on multiple servers.
- Applications/services managed the data and resources
  Corporate applications and services allow access to the data and resources of the corporation. No special tools were used to monitor their use.

Each layer in the "defense-in-depth" approach required rights, permissions and expertise to allow a compromise to occur. This corporation felt their environment was secured. We demonstrated to them that they could not systematically quantify or identify potential threats from the outside. The corporation was at grave risk of intrusion and the security baseline as implemented was not able to identify that risk.

Firewalls and router were simply not the complete answer to their security. We started by assisting the corporation in identifying threats and vulnerabilities. Later, we demonstrated mechanisms that would allow attackers to breach their environment in a systematic way.

Threats and vulnerabilities

Before these outside threats could be quantified, this corporation needed to understand how to identify and quantify threats and vulnerabilities. Known threats

and vulnerabilities are documented in security literature or on various security-related web sites such as http://icat.nist.gov/icat.cfm (searchable index of computer vulnerabilities) or http://cve.mitre.org (vulnerability standards site). These various sources can be used to identify and quantify the threats and vulnerabilities that concern a particular corporation. Internal source of information can augment the industry specific knowledge.

The composition of an attack as defined in the Hacking Exposed: Network Security Secrets & Solutions (Second Edition) security book[1] is divided into several categories:

1. *Foot printing* is a non-intrusive process of collecting information about a corporation's network and resources. Open searches on the Internet will reveal much of this detailed information.

2. *Scanning* can be performed passively by listening to traffic on the network or actively by probing ports on a host machine[2]. Scanning can detect potential weaknesses in a corporate network.

3. *Enumeration* is the process of extracting information from a particular host and usually concerns information about network resources, shares, groups, users and applications.

4. *Gaining Access* is the phase in which a host machine is breached. The attacker normally acquires restricted access but some intrusion mechanisms allow privileged access immediately.

   An attacker often leaves behind a means to ensure easy access the next time. Sometimes this includes simply leaving a hard to find unauthorized user name and password. Other times the attacker leaves behind stealthy remote control software called a backdoor.

5. *Escalating Privileges* allows the attacker to gain system level access (if not already obtained) on a host machine and possibly assume corporate-wide administrative status within the entire corporate environment.

6. *Pilfering* is the act of compromising corporate resources or data. Intellectual theft of corporate data is a real potential.

7. *Covering Tracks* allows the attacker to remove any information from the host(s) that may indicate an intrusion has occurred.

[1] Scambray, chapters 1,2,3,5,6
[2] Cole, p. 25-26.

All the steps in this list are important. We choose to concentrate on the first 5 steps (foot printing, scanning, enumeration, gaining access and escalating privileges) to demonstrate to the aforementioned corporation the extent of their security exposure.

Attack Tree Analysis

This corporation agreed that a structured approach was necessary to better understand any outside threats arrayed against it. Various methods were available but the attack tree methodology was suggested as the most complete and comprehensive to solve their problem. In short, the attack tree methodology was chosen because it identified threats in a structured way and easily exposed countermeasures. The staff at this corporation was experiencing difficult in identifying threats in relationships to their infrastructure and applications. The attack tree methodology allowed for the decomposition of threats as identified and was easy to implement. Countermeasures are easily related to the identified threats in the context of this methodology. Successful attack correlations require that exact countermeasures to be identified.

In general, the attack tree methodology presents a mechanism to identify specific threats and detail them in a hierarchical manner. Countermeasures are detailed at each level of the tree when deemed appropriate and specify the means to detect or prevent a threat(s).

The tree designer starts with a high level goal that defines the overall threat. A simple “AND/OR” paradigm is followed until the bottom leaves of the tree are reached. Each higher-level leaf in the tree is decomposed until all appropriate threats are defined. The attack tree methodology demands that a decision is made at each level. An AND condition requires that all nodes at the same level are performed in order to transcend to the next level. An OR condition only requires a single node at the same level to be performed in order to reach the next level.

The attack tree methodology gives the designer the ability to isolate threats. The manner and order in which each threat is realized is taken care of by the attack tree methodology. For example, if someone wishes to read a newspaper, they can purchase it, steal it, lend it from another or simply go without reading it – an OR condition. If they decide to purchase the newspaper, several steps are required to transpire in a serial manner: find a newspaper vending machine, insert the correct amount of coins, open the vending machine’s door and take the newspaper – an AND condition. There are certainly other means to purchase a newspaper but you get the idea. The means of acquiring a newspaper allow for many possible sequences to occur (OR). If we decide to purchase a newspaper legitimately from a vending machine, a series of steps ensue (AND).

Threats are easier to prevent with an AND sequences. If we decide to purchase a newspaper legitimately from a vending machine and we have no money, the ability to legitimately open the vending machine's door is very limited. On the other hand, we could borrow the newspaper from another or steal it if we have no money. Hence, the OR condition is more difficult to defense[3].

Attack Tree Scenarios

The attack tree methodology allowed this corporation to quantify their exposure to the real world. Their infrastructure and business processes were maintained exclusively on Intel-based hardware and utilized the Windows 2000 operating system. The major applications allowed to communicate with the outside world were Exchange 2000 and IIS 5.0. The scenarios in this paper are restricted to their Windows 2000 and IIS-related vulnerabilities.

In general, attack tree scenarios must take into account the ability of an attacker to identify multiple means to breach a corporate environment. Attackers recognize the complexity of an enterprise and normally search for the easiest way to conduct an attack. A badly configured web-server without the appropriate security patches is a likely candidate.

Attacks are usually mounted incrementally and begin by gathering intelligence. The attacker first footprints a corporation and then scans its networks for weaknesses such as open ports or active services running on a host machine. Enumeration, the final step in the intelligence-gathering process, supplies the attacker with the information to remotely gain access to a corporate host machine(s).

The access to an internal computer allows the attacker to escalate privileges on that machine. The security breach on a single computer could and most likely will allows the attacker to gain the ultimate prize: corporate-wide administrative privileges. At this point, the informational assets and intellectual property of the enterprise are simply to be plucked.

Attack tree scenarios can identify, quantify and suggest countermeasures to detect and possibly prevent these intrusions. The attack tree scenarios in this paper will concentrate on two of the bedrock principles of security: confidentiality and integrity. Availability can be added later.

[3] Moore, Andrew, Ellison, Robert. "Survivability through Intrusion Aware Design", p. 3.

This is the strength in the hierarchical nature of the attack tree methodology. Different sections of the tree can be designed, refined or enhanced separately. In fact, different groups within the corporate structure can be assigned responsibility for a section of the tree without affecting another group. Each section of the tree gravitates to the group with the expertise to manage that specific section[4].

This paper covers the following attack tree scenarios:

- Gather intelligence (Confidentiality)
- Gain remote access and escalate privileges (Confidentiality/Integrity)

Gather Intelligence

Figure 1 defines the steps required to gather intelligence about the corporate environment. This attack tree scenario concentrates specifically on the Windows 2000 environment due to the infrastructure makeup of the aforementioned corporation. Obviously, this is only a subset of the tools available to perform intelligence gathering. Others can be added later. But it does give us a good cross-section of the techniques required to complete the intelligence gathering steps.

The AND condition on node 1 indicates that the intelligence gathering step must be completed in order to proceed to next level of the tree defined in Figure 2. All the intelligence gathering steps – foot printing (node 1.1), scanning (node1.2) and enumeration (node 1.3) must be accomplished serially – hence the AND conditions. Various means maybe implemented to complete the foot printing and scanning nodes – hence the OR conditions on each sub-node. Enumeration (node 1.3) requires that share (1.3.1) and user (1.3.2) information be collected to complete this process. Different options are available to collect both user and share information however.

Countermeasures are included when appropriate. For example, the countermeasure for the *Scanning* is to monitor the corporate network with a detection product(s).

All nodes in this tree are detailed below Figure 1. Countermeasures for the *gather intelligence* attack tree are described in the last sub-section: **Countermeasures**.

[4] Moberg, p. 21.

Figure 1 Attack Tree – gather intelligence (Windows 2000 Environment)

Goal: gathering intelligence
1. Gather intelligence (AND)
   1.1. Foot printing (AND)
      1.1.1. DNS zone transfer (OR)

      Disallow the ability to perform zone transfers or block zone transfer information leaving the corporate environment at the firewall. Monitor the DNS zone transfer activities with Network Intrusion Detection System (NIDS) and Host Intrusion Detection (HIDS) products.

   1.2. Scanning (AND)
      1.2.1. Use FSCAN (OR)
      1.2.2. Use ipEye (OR)

      Monitor the corporate network with NIDS product.

   1.3. Enumerate (AND)
      1.3.1. Shares (AND)
         1.3.1.1. Perform Null sessions (OR)
         1.3.1.2. Use DumpAcl (OR)
      1.3.2. Users (AND)
         1.3.2.1. Use enum (OR)
         1.3.2.2. Use DumpAcl (OR)

      Monitor foreign tool usage with HIDS and System Integrity Verifiers (SIV).

Here is a brief explanation of the individual nodes of the attack tree shown in Figure 1:

1. Gather intelligence - The standard intelligent gathering techniques (foot printing, scanning and enumeration) must be accomplished in order to gain access.

1.1 Foot printing – Foot printing of a corporate Domain Name Servers (DNS) is usually the last step in the foot printing. The majority of foot printing is accomplished by querying open source databases on the web that usually supply much information about a corporation.

1.1.1 DNS zone transfer – DNS is the distributed database for name or IP address resolution and can supply a lot of information about a potential corporation.

A simple zone transfer (nslookup -d) could be performed on the primary DNS server that could garner vital domain information. Zone transfers are performed between the primary and secondary DNS servers to exchange and synchronize information. The zone transfer should be disabled or minimally the transfer of DNS information should be blocked to the outside.

nslookup –d

```
> ls -d hub.netiq.local
[localhost]
 hub.netiq.local.            SOA    boston.hub.netiq.local admin.hub.netiq.lo
cal. (1 900 600 86400 3600)
 hub.netiq.local.            NS     boston.hub.netiq.local
 boston                      A      192.168.1.10
 hub.netiq.local.            SOA    boston.hub.netiq.local admin.hub.netiq.lo
cal. (1 900 600 86400 3600)
```

This particular “nslookup” command returns the DNS server name, IP address and potential administrator’s name.

1.2 Scanning – Scanning involves collecting information about a network that includes active ports, type of operating system and services running. Passively listening to network traffic or actively probing a machine can gather this information.

1.2.1 FScan – FScan is a command line port scanner that scans both TCP and UDP ports. More details about FScan can be found at http://www.foundstone.com/knowledge/proddesc/fscan.html. The example scan below is scanning ports 20-100 on a remote host. Port 27/tcp was found to be active.

```
D:\>FScan.exe -p 20-100 192.168

FScan v1.12 - Command line port scanner.
Copyright 2000 (c) by Foundstone, Inc.
http://www.foundstone.com

 Scan started at Sat Jun 15 16:20:25 2002

192.168.1.11       27/tcp

 Scan finished at Sat Jun 15 16:20:27 2002
 Time taken: 81 ports in 1.261 secs (64.23 ports/sec)
```

1.2.2 use ipEye – ipEye is a port scanner for the Windows environment that can do SYN, Null and Xmas scans. More details about ipEye can be found at http://ntsecurity.nu/toolbox/ipeye/. The example scan below is scanning ports 20-100 on a remote host. Port 27/tcp was found to be active.

```
D:\>ipeye.exe 192.168.1.11 -syn –p 20 100

ipEye 1.1 - (c) 2000, Arne Vidstrom (arne.vidstrom@ntsecurity.nu)
        - http://ntsecurity.nu/toolbox/ipeye/

  1-19 [not scanned]
  20-26 [closed or reject]
  27 [open]
  28-100 [closed or reject]
  101-65535 [not scanned]
```

1.3 Enumerate – Enumeration is the process of actively collecting information on a host machine.

1.3.1 Shares – Collecting information on shares facilitates access to those shares.

1.3.1.1 Perform Null sessions – Null Sessions is the ability to access a Windows 2000 share with a blank user name and password. An attacker simply uses the net use command to gain access to an unprotected share as shown below:

```
C:\>net use \\192.168.1.10\IPC$ "" /u:""
The command completed successfully.
```

1.3.1.2 Use DumpAcl – DumpAcl is a graphical based enumeration tool that allows for the collection of share, user and group information remotely. More details about DumpAcl can be found at http://www.mvps.org/win32/security/dumpacl.html. The enumeration example below utilizes the command line interface from DumpAcl to dump share information from a remote host.

```
Dumpacl /rpt=shares outfile=d:/share.dcl /computer=boston

6/15/02 1:26 PM - Somarsoft DumpAcl - \\boston
Path (exception dirs and files)      Account        Own Dir        File

\\boston\ipc$= (special admin share)              admin-only (no dacl)

\\boston\ipc$\          ==>access denied
```

1.3.2 Users – Collecting information on user facilitate access to those users.

1.3.2.1 Use enum – enum is a console-based Win32 information enumeration utility. It utilizes null sessions to retrieve lists of users, machines, shares, names, groups and members and passwords. enum is also capable of a rudimentary brute force dictionary attack on individual accounts. More details about enum can be found at http://razor.bindview.com/tools/index.shtml. The enumeration example below utilizes the command line interface from enum to dump detailed user information from a remote host.

```
enum -U -d 192.168.1.10

server: 192.168.1.10
connected as NEARBOSTON\netiq, disconnecting... success.
setting up session... success.
getting user list (pass 1, index 0)... success, got 8.
  __vmware_user__ (VMware User)
  attributes:
  Administrator (Built-in account for administering the computer/domain)
  attributes:
Guest (Built-in account for guest access to the computer/domain)
  attributes: disabled no_passwd
  IUSR_BOSTON (Built-in account for anonymous access to Internet Information
Services)
  attributes: no_passwd
  netiq   attributes:
cleaning up... success.
```

1.3.2.2 Use DumpAcl – DumpAcl is a graphical based enumeration tool that allows for the collection of share, user and group information remotely. More details about DumpAcl can be found at http://www.mvps.org/win32/security/dumpacl.html. The enumeration example below utilizes the command line interface from DumpAcl to dump user information remotely.

```
Dumpacl /rpt=usersonly outfile=d:/users.dcl /computer=boston

6/15/02 3:26 PM - Somarsoft DumpAcl - \\boston
UserName          FullName      Comment
Administrator     Built-in account for administering the computer/domain
Guest             Built-in account for guest access to the computer/domain
IUSR_BOSTON       Internet Guest Account
IWAM_BOSTON       Launch IIS Process Account      Built-in account for IIS
krbtgt            Key Distribution Center Service Account
```

Gain Remote Access and Escalate Privileges

The intelligence gathering steps must first be completed before we can attempt to define the ability to remotely gain access to the Windows 2000 host machine(s). Later, privilege escalation could be sought to demonstrate the ability to pilfer corporate data and resources.

Figure 2 defines the steps required to gain remote access, escalate privileges on a corporate host machine(s) and leave undetected. This attack tree scenario concentrates specifically on the Windows 2000 environment due to the nature of the corporate infrastructure in question. Obviously, this is only a subset of the tools and mechanisms available to gain remote access, escalate privileges and leave without detection. Others can be added later. But it does give us a good cross-section of the techniques required to complete these steps.

Gaining remote access and escalating privileges require that the intelligence gathering steps were completed – hence an AND condition on node 2. The attacker must gain access remotely before the next steps can occur – *escalate privileges* and *exit without detection*. This is demonstrated by the AND condition in nodes 2.1 through 2.3. There are several means to accomplish access remotely – see nodes 2.1.1 through 2.1.3. The next step "*escalate privileges*" can be accomplished in many ways but only a LPC exploit is detailed. Other may be added. The last step is to exit the machine without detection. The attacker has the option to leave behind a backdoor mechanism to easily regain access at the privilege level already obtained. Obviously, the attacker wishes to maintain the escalated privileges gained by the intrusion and for this reason should not be caught. Otherwise, the privileges would be revoked and any backdoor mechanism would be removed.

All nodes in this scenario are detailed below Figure 2. Countermeasures for the *gain remote access and escalate privileges* attack tree scenarios are described in the last sub-section: **Countermeasures**.

Figure 2 Attack Tree – Gain remote access and escalate privileges (Windows 2000 Environment)

Goal: Gain access to corporate resources and escalate privileges
2. Gain remote access and escalate privileges (AND)
    2.1. Gain access remotely (AND)
        2.1.1. Guess password (OR)
            2.1.1.1. Use native command: net use (OR)
            2.1.1.2. Use Brutus (OR)
        2.1.2. Eavesdrop on network for password (OR)

2.1.2.1. Use LC4 (OR)
2.1.2.2. Use WinDump (OR)
2.1.3. Perform exploit (OR)
2.1.3.1. Perform Unchecked Buffer In ISAPI Extension Compromise (OR)

Monitor the network with a Network Intrusion Detection System (NIDS) and host machines with a Host Intrusion Detection System (HIDS). Implement strong password policy.

2.2. Escalate Privileges (AND)
2.2.1. Perform exploit (OR)
2.2.1.1. Perform LPC Port System Call (OR)

Monitor host machine with Vulnerability Assessor (VA) and HIDS. Apply proper security patches. Harden servers/workstations and critical business applications.

2.3. Exit without detection (AND)
2.3.1. Leave a backdoor (OR)
2.3.1.1. Use BO2K (OR)

Monitor network with NIDS and host machine with Anti-Virus, VA and HIDS.

Here is a brief explanation of the individual nodes of the attack tree shown in Figure 2:

2. Gain remote access and escalate privileges – The aim is to gain access to a host machine and escalate privileges to allow the access to corporate data and resources. The last step is to leave undetected.

2.1 Gain access remotely – The aim is to gain access remotely by password guessing, eavesdropping on the network or performing an exploit.

2.1.1 Guess password – The ability to guess or crack a password allows the attacker to logon in remotely as a legitimate user.

2.1.1.1 Use native command – The command *net use* certainly allows the attacker a simple mechanism to guess passwords from the command line as shown below:

```
D:\>net use \\192.168.1.10\IPC$ * /user:administrator
Type the password for \\192.168.1.10\IPC$:
The command completed successfully.
```

2.1.1.2 Use Brutus – Brutus is a graphical-based remote on-line password cracker and allows for dictionary, hybrid or brute force password guessing. More details about Brutus can be found at http://www.hoobie.net/brutus/.

2.1.2 Eavesdrop on network for password – The ability to capture a user's id and password as it travels over the network rather than authenticating or breaching a server or workstation.

2.1.2.1 Use LC4 – LC4 is a graphical-based password cracker and allows for dictionary, hybrid or brute force password guessing and is the most powerful and popular password-guessing tool in the Windows environment. LC4 can also sniff passwords from the network. More details about LC4 can be found at http://www.atstake.com.

Here is some output from LC4 detailing users and their passwords:

USERNAME	LANMAN PASSWORD	NTLM PASSWORD
aagassi	* empty *	* empty *
aaldridge	* empty *	* empty *
aale	AALE	aale
aamaya	* empty *	* empty *
aambrose	* empty *	* empty *
abeasley	* empty *	* empty *
aberasategui	* empty *	* empty *
aberg	ABERG	aberg

2.1.2.2 Use WinDump - WinDump is a packet sniffer that allows the user to passively capture information over the wire. More details about WinDump can be found at http://windump.polito.it/.

2.1.3 Perform exploit – Any number of vulnerabilities can be attacked on any given machine that may result in obtaining user or administrative privileges. If administrative privileges are acquired, privileged escalation has already occurred. The AND condition in node 2.2 is already accomplished.

2.1.3.1 Unchecked Buffer In ISAPI Extension Compromise – This is an example of a newly discovered buffer overflow attack in IIS 5.0.

"A buffer overflow occurs when a program or process tries to store more data in a **buffer** (temporary data storage area) than it was intended to hold. Since buffers are created to contain a finite amount of data, the extra information - which has to go somewhere - can overflow into adjacent buffers, corrupting or overwriting the valid data held in them. Although it may occur accidentally through programming error, buffer overflow is an increasingly common type of security attack on **data integrity**. In buffer overflow attacks, the extra data may contain codes designed to trigger specific actions, in effect sending new **instruction**s to the attacked computer that could, for example, damage the user's files, change data, or disclose confidential information.

Buffer overflow attacks are said to have arisen because the **C** programming language supplied the framework, and poor programming practices supplied the vulnerability. [5]"

Buffer overflow attacks against an IIS server are difficult to prevent because the exploit occurs using legitimate web ports such as port 80 (HTTP). If data is passed through a legitimate web port and corrupts a particular buffer in the web service, malicious code can be introduced. Windows 2000 restarts the web service by default thus allowing the malicious code to run in the context of the newly started web service. The web service normally runs in the context of the local system account that allows access to all resources and data on that web server machine.

The vulnerability is well documented on the ICAT vulnerability web site under: http://icat.nist.gov/icat.cfm?cvename=CVE-2001-0241.

2.2 Escalate Privileges – The major objective in breaching a host machine is to increase system privileges that allow pilfering of corporate data and resources.

2.2.1 Perform exploit – Any number of vulnerabilities can be attacked on any given machine that may result in acquiring escalated privileges.

2.2.1.1 Perform LPC Port System Call – The Local Procedure Call allows efficient communication for processes on a local machine. Any process that knows the process, thread and message identifiers of an LPC message can access it. The identifiers are predictable. Local privilege escalation is possible. The LPC port request may also be spoofed allowing an attacker to create a process that runs in the context of the local system process. The local system account allows access to all resources and data on the machine.

The vulnerability is well documented on the security focus web site: http://online.securityfocus.com/archive/1/137347.

2.3 Exit before detection – The attacker needs to logoff the machine undiscovered. If the attacker is discovered, access and privileged escalation are lost.

2.3.1 Leave backdoor – An attacker does not want to lose the privileged access already acquired. Usually a remote administrative tool or unauthorized account is stealthy left behind to easily regain access.

---

[5] Kramer, David. SearchSecurity.com Definitions: Buffer overflow. 05 May 2001. URL: http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci549024,00.html (25 June 2002).

2.3.1.1 Use BO2K – BO2K or Back Orifice 2000 is a stealthy remote control tool. See http://sourceforge.net/projects/bo2k/ for more information.

Countermeasures

Countermeasures really start with defense in depth. In general, a corporation has to institute a layered defense that usually includes a multitude of perimeter (firewalls, routers) and internal defenses (hardening O/S, installing latest security patches and utilizing detection/security tools).

Firewalls and routers perform a vital security process. However, attackers can "sneak around them or go through them to avoid any filtering"[6]. Most web servers, for example, accept unfiltered traffic to port 80.

A strong password policy ensures that passwords are not easily guessed or cracked. Microsoft (http://www.microsoft.com) has many articles on their web site discussing strong password policy implementation in Windows 2000 and should be consulted.

Hardening or securing of Windows 2000 servers or workstations reduces their exposure to potential attacks by eliminating known security loopholes. See **Hardening Windows 2000 in the Enterprise Part One: Seeing the Forest in Spite of the Trees** white paper at http://online.securityfocus.com/infocus/1296 as an example of a hardening guideline.

The hardening of critical business application should be taken on a case-by-case basic. The security industry has developed strict guidelines on hardening applications. Security experts, literature and web sites can be consulted to develop an appropriate *hardening* guideline to meet the security objectives of the corporation in question.

Attacks and exploits occur daily. Corporations are forewarned to install the latest security patches. Vendors constantly post their security patches to the web and security related sites such as http://cve.mitre.org track the latest security patches by vulnerability.

The practice of hardening, applying security patches and implementing a strong perimeter defense with firewalls and routers is not enough. Corporations today must proactively detect and protect against intrusions with a set of detection products and security tools. Each category of security tools and detection

[6] Schneier, Bruce. Secrets and Lies. New York: John Wiley &Sons, Inc., 2000. p. 190

products identifies specific security problems and these categories are described as follows[7]:

- Network Intrusion Detection System (NIDS) – examines network traffic and looks for abnormalities in network packages based on signatures and heuristics.
- System Integrity Verifiers (SIV) – ensures that critical system and corporate files have not been compromised through the use of checksums and hashing mechanisms.
- Vulnerability Assessors (VA) – statically checks a computer for known vulnerabilities and suggests fixes. Their strength derives from their ability to scan for most known vulnerabilities. These scan can take time depending on the level of the scan and are usually scheduled.
- Anti-Virus (AV) – scans a computer in relation to the aforementioned scenarios for backdoor products.
- Log File Consolidator (LFC) – collects operating system and applications logs and consolidates them into a single source.
- Host Intrusion Detection System (HIDS) – examines standard host-based source information like syslogs looking for possible intrusions. HIDS are often rule-based and some allow for event correlation.

The complexity of correlating the security incidences in the simplified attack tree scenarios described in Figure 1 and 2 is overwhelming. Each category of detection system product or security tool has its strength and weaknesses in detecting and notifying us about a specific instance of an attack.

HIDS can tell us that a particular hacker tool is running. SIVs can tell us that certain critical system files are compromised. NIDS can tell us that abnormalities are occurring on the network. All these detection systems report certain categories of incidences in a detailed manner.

How do we better correlate the myriad of security incidences with which we are confronted? How do we deliver knowledge to detect and possible prevent correlated attacks?

Attack Correlation Workflow

The corporation in question lives with risks everyday. The attack tree methodology helped to identify and manage that risk. If other corporations today are going to manage that risk, they too need to follow a structured approach with a methodology like attack tree to quantify it. The threat can then be detected and hopefully prevented.

[7] Walker, p. 4.

Attack correlation becomes another means of implementing the "defense in depth" approach and is just another countermeasure albeit a very powerful one. Attack correlations require a workflow to identify and define correlations.

Fredrick Moberg performed an analysis at Volvo that defined a workflow for defining attack tree scenarios[8]. This paper proposes a workflow to define attack correlations within the confines of that same attack tree methodology.

The workflow should include the following steps:

1. Define/refine attack tree scenarios
2. Correlate attack
3. Maintain attack knowledge base
4. Verify correlation

Define/Refine Attack Tree Scenarios

Attack tree scenarios are constructed to define the threats that could place a corporation at risk. This paper already defined the process of identifying threats and creating attack tree scenarios. Attack tree scenarios present the corporation with a structured approach to better understand these threats and the countermeasures necessary to detect if not prevent them.

Corporations can use the attack tree methodology to define new threats or refine existing ones. The structured approach allows a corporation to understand and define possible attack correlations. For example, unusually heavy port scanning activity on a web server that is followed by stopping and starting of the web service may indicate a potential buffer overflow attack (see 2.1.3.1 Perform Unchecked Buffer In ISAPI Extension Compromise).

Correlate Attack

Every category of detection product or security tool can analyze a complex incidence and return a set of discrete events. Normalization of a complex security incidence is defined by these discrete events. These discrete events are normally stored in a system or an application-specific log. A Log File Consolidator (LFC) can be used to consolidate these discrete events disturbed on various servers and workstations throughout the corporation.

Consolidation of these discrete events allows for the centralization of the corporate-wide incident handling system. A Host Intrusion Detection System (HIDS) normally encapsulate a LFC and use the collected events to apply their

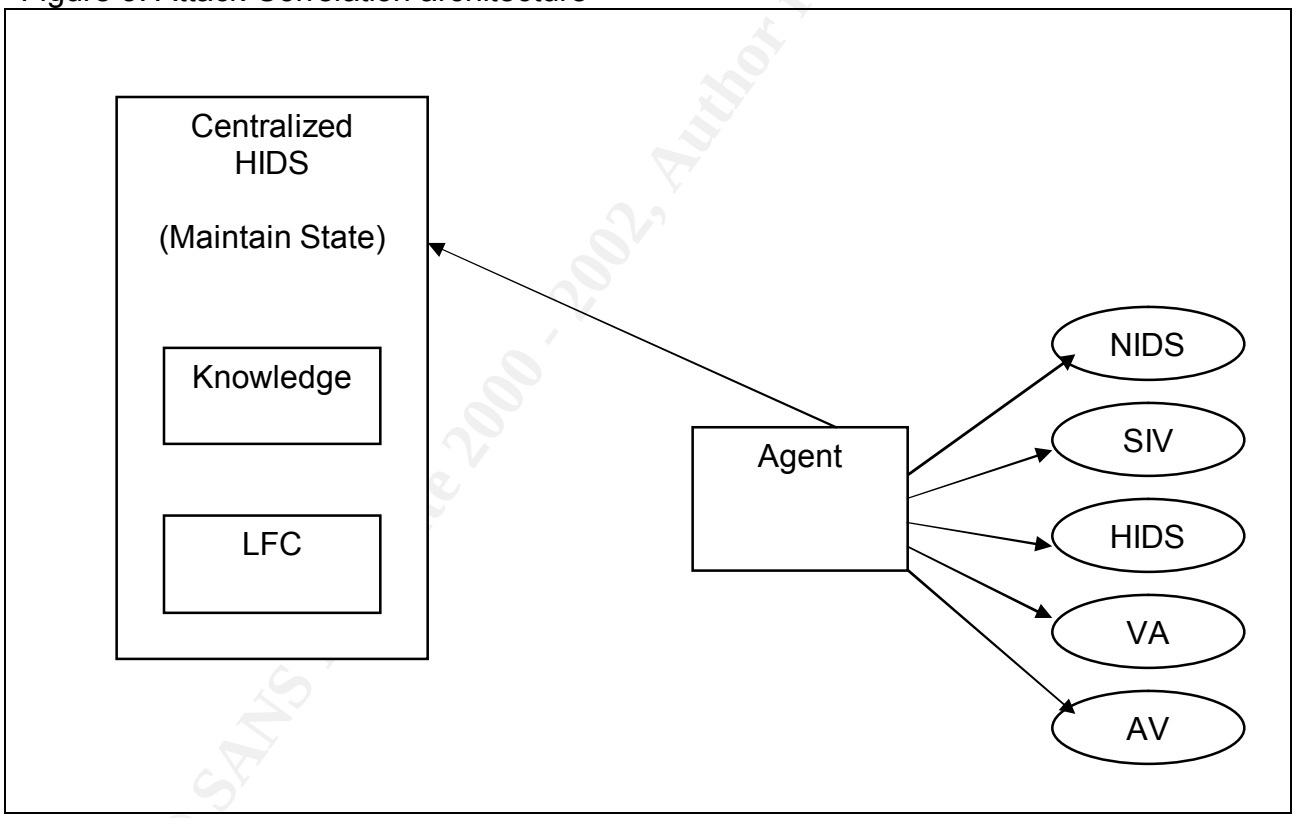
[8] Moberg, p. 21.

detection rules. These rules allow the HIDS to detect any security incidence. These rules can be related directly to nodes in an attack tree scenario.

Consolidation is the initial step towards correlation. The consolidation of the discrete events allows for correlation rules to be applied at an event level. Attack correlation can then be achieved with the technology in the marketplace today.

The architecture is based on agent technology (See Figure 3). The agent receives sets of rules from the centralized HIDS. These sets of rules direct the agent to collect event information from the diverse set of security tools or products such as a NIDS running on a remote machine. The discrete events are collected and pushed to the centralized HIDS server.

Figure 3. Attack Correlation architecture



The corporate-wide attack tree hierarchy allows us to identify possible correlation points. For example, port scanning and the logging of frequent web services "*stop and start*" activity could imply a more sophisticated attack (i.e. access a host machine remotely) rather than just a set of unrelated security incidences. Correlation of these incidences will prove valuable in understanding and preventing these types of more sophisticated attacks.

In short, the individual nodes in an attack tree scenario can show correlations that may indicate a more sophisticated attack. The centralized HIDS captures each of these incidences (i.e. as defined by the attack tree nodes) as discrete events. The "state" of these individual incidences can then be maintained by the centralized HIDS. These state variables can include such sophisticated information as time stamps, source IP addresses or port numbers.

For example, a firewall (log analyzed by the HIDS) and a NIDS (placed outside the firewall) could collect distributed port scanning information. This would entail collecting the frequency of the scan, the source/target IP addresses and the target port number. If a single source address is targeting (1) multiple scans on a single port and (2) scans on the same port number across multiple machines within a certain time frame, this correlation can be demonstrated. This may indicate that further attacks such as a buffer overflow exploit are eminent.

The centralized HIDS performs the following actions:

1. maintains a state matrix of correlated incidences,
2. collects the discrete events from the detection products and security tools in-place (see Figure 3),
3. normalizes these events and updates their state in the matrix,
4. analyze the matrix periodically looking for correlations and
5. generate an alert if a correlated attack is in progress or has occurred.

The attack tree scenario contains nodes that correspond to the individual incidences. The structured nature of the attack tree scenario allows for the identification of correlations that might otherwise go unseen.

Maintain Attack Knowledge Base

Knowledge from a set of security incidences can then be linked together to supply a more sophisticated view of an attack and not just a single incidence. Security knowledge is then supplied by the centralized HIDS system to the organization at the level required.

The knowledge of the individual incidences is constantly updated as the attack tree is actively maintained. This knowledge of each of these incidences is feed into the centralized HIDS that correlates it to the individual rules. The rules or set of rules would have linkage to the individual nodes in an attack tree scenario.

The knowledge assigned to these rules would be accumulated to match the attack correlations maintained by the centralized HIDS. The security personnel within the corporation would then be supplied with information on the correlated attack rather than just information on a set of unrelated incidences. This would be

very beneficial to research and fix potential security loopholes in the corporate infrastructure.

Verify Correlation

The attack can then be documented and analyzed. The analysis would corroborate the attack correlation(s) or require its redefinition. It may also demand that the attack tree scenario requires redefinition. A new attack tree scenario maybe required. The analysis insures that the correlation and the attack tree scenario remain updated to meet the ever-changing nature of threats and vulnerabilities.

Outcome

The corporation in question had done its do diligence and produced the attack scenarios as described in the *gather intelligence* (Figure 1) and *gain remote access and escalate privileges* (Figure 2). The management at this corporation now understood the risks involved and instituted a plan to implement the suggested countermeasures. The layout of the countermeasure were implemented as follows:

- DNS zone transfers were disallowed.
- All critical application servers and applications were hardened, security patches applied and a strong password policy implemented.
- The configuration of the firewalls and the routers were configured upon agreed on corporate guidelines.
- A Host Intrusion Detection System (HIDS) was installed on all critical servers and workstations.
- The centralized HIDS was configured for attack correlation detection.
- A Network Intrusion Detection System (NIDS) was installed to monitor traffic on the corporate network.
- Vulnerability Assessor (VA) software was scheduled to generate static vulnerabilities on all business critical servers.
- Anti-Virus (AV) software events were captured.
- System Integrity Verifiers (SIV) software was installed to ensure the integrity of critical system files.
- All required discrete events were forwarded to the centralized HIDS.

The attack tree methodology allowed the corporation to better understand their risks and promoted the adoption of countermeasures.

Attack correlations were the last countermeasure implemented. This corporation was concerned that port scanning would be used extensively to allow enumeration and buffer overflow attacks to occur. The second major concern was that a clever set of outside attackers could by-pass the countermeasures in-

place. If the corporation was indeed penetrated, the detection of privilege escalation and any backdoors is essential.

Three attack correlation detection scenarios were implemented:

1. Port scanning-Enumeration

   Port scanning-Enumeration starts by verifying that attackers were randomly targeting particular set of ports on multiple machines while repetitively scanning theses same ports on a particular machine. For example, active web ports 80 (HTTP), 21 (FTP) and 443 (SSL) would most likely indicate a web server. Attackers would scan for these active ports against all machines on the corporate network perimeter and then intensively scan those same ports on a particular set of (most likely web) servers. The ability to correlate this activity with attempts to enumerate information via tools such as enum (see attack tree node 1.3.2.1 in Figure 1) certainly would be cause for alarm. The correlated discrete events from the HIDS, NIDS and SIV detection tools could produce an alert in such an eventuality.

2. Port scanning-Buffer Overflow

   Port scanning-Buffer Overflow would be a variation of the previous one. This correlation would certainly be more difficult to detect but would be possible. The discrete events from the port scanning activity as described above would be correlated with the unauthorized stopping and starting of the web service and the possible introduction of a buffer overflow generated process (see attack tree node 2.1.3.1 in Figure 2). The HIDS and NIDS detection products could produce an alert in such an eventuality.

3. Privilege escalation-Backdoor

   Privilege escalation-Backdoor would verify that a LPC port request was spoofed to allow privileged escalation to occur. The HIDS could verify that a LPC generated process as well as a backdoor mechanism such as BO2K were running. The NIDS could detect data streams to an unauthorized host. The AV tool could certainly detect a backdoor mechanism on a server. A combination of discrete events from the various detection products and security tools would be utilized to implement this correlation.

These correlations did assist this corporation in identifying and preventing potential attacks. Intensive port scanning always seemed to follow web server and enumeration attacks. Privilege escalation attacks were identified, reversed and all known backdoors were eliminated. The corporation was no longer left waiting for potential threats to occur but had the mechanism in-place to identify and potential stop a correlated attack.

Were all threats eliminated? Not likely, but security awareness has risen in the corporation markedly. This process did require the corporation to analyze threats constantly. Security is a process that must always be reviewed. Today, the corporation must implement their own correlations. Certainly, it would be advantageous for the centralized HIDS to come with a set of correlation rules built-in. Even more advantageous would be an attack tree scenario editor that would help define scenarios and allows for the definition of correlations. The attack tree editor could then generate the appropriate rule sets to identify threats and the "state" matrix to ensure the correlation.

The payoff comes from the identification of potential correlated attacks. The corporation also no longer needs to depend on security experts to detect and possibly prevent security breaches. Second-level support personnel can now be utilized. The centralized HIDS not only detects potential correlated attacks but also supplies the knowledge to understand them. The corporation can now reserve their senior security experts to work on more compelling issues and allow less experienced personnel to perform the "fire fighting" role. These second level support personnel now have the appropriate knowledge to work on these more complex issues.

The one big issue with the definition of attack correlations was the issue of "false positives". Correlations do depend on the discrete events from different security and detection tools. If these tools incorrectly generate an event indicating "abnormal" behavior but this behavior is later diagnosed as normal, a "false positive" has occurred. Certainly, a NIDS can detect "abnormal" network traffic that is later found to be normal. A HIDS can falsely indicate an IIS-related vulnerability in interpreting a valid web service as compromised. Correlation tuning must be accomplished to mitigate "false positives". For example, this tuning can be accomplished by verifying that certain software patches are missing on a machine that would allow the threat to occur in the first place.

Attack correlations work in the context of a structured method such as the attack tree methodology. Neither the attack tree methodology nor attack correlations can eliminate all threats and their associated vulnerabilities but these methods can be used to better understand and detect them. New threats are easier to quantify because of the structured approach and can be quickly added to an attack tree scenario. A new correlation can possibly be identified and defined. Risk is not eliminated but certainly quantified.

The effort was certainly worth it. Correlations are a powerful countermeasure. Correlations should be implemented incrementally and as experience is gained more sophisticated correlations can be added.

Conclusion

The process of decomposing and refining the threats in a hierarchical manner reveals the countermeasures required. The deployment of countermeasures ensures a structured, stable environment that allows for the implementation of reliable and accurate attack correlations. These correlations allow the corporation to detect sophisticated attacks that imply more than a single incidence.

Attack correlations are a powerful countermeasure and can be implemented with the security technology today. Corporations are advised to use a methodology such as attack tree to better define threats against them. The structured definition of these threats will allow them to identify and implement attack correlations. These attack correlation could just prevent an embarrassing compromise.

References

Cole, Eric. Hacker Beware. Indianapolis: New Rider, 2002.

Moberg, Fredrick. "Security of an Information System using an attack tree methodology". 16 November 2000.
URL: http://www.ce.chalmers.se/staff/jonsson/fredrik.moberg-thesis.pdf (16 May 2002).

Moore, Andrew, Ellison, Robert, Linger, Richard. "Attack Modeling for Informational Security and Survivability". CMU/SEI –2001-TN-001. March 2001.
URL: http://www.cert.org/archive/pdf/01tn001.pdf (16 May 2002).

Moore, Andrew, Ellison, Robert. "Survivability through Intrusion Aware Design". 28 August 2001. URL: http://www.cert.org/research/isw/isw2001/papers/Moore-28-08.pdf (16 May 2002).

Mullen, Timothy. "Hardening Windows 2000, Part One: Seeing the Forest In Spite of the Trees". 21 May 2001.
URL: http://online.securityfocus.com/infocus/1296 (1 June 2002).

Scambray, Joel, McClure Stuart and Kurtz George. Hacking Exposed: Network Security Secrets & Solutions (Second Edition). Berkley: McGraw Hill, 2001.

Schneier, Bruce. "Why Cryptography Is Harder Than It Looks". CounterPane Systems White paper. 1997. URL: http://www.counterpane.com/whycrypto.html (16 May 2002).

Schneier, Bruce. "Modeling security threats". *Dr. Dobb's Journal* December 1999.
URL: http://www.counterpane.com/attacktrees-ddj-ft.html (16 May 2002).

Schneier, Bruce. Secrets and Lies. New York: John Wiley &Sons, Inc., 2000.

Schumacher, Markus and Roedig, Utz. "Security Engineering with Patterns". 27 July 2001. URL:
http://jerry.cs.uiuc.edu/~plop/plop2001/accepted_submissions/PLoP2001/mschumacher0/PLoP2001_mschumacher0_1.pdf (16 May 2002).

Steffan, Jan and Schumacher, Markus. "Collaborative Attack Modeling". ACM 1-58113-445-02/02/03. 2 February 2002.
URL: http://www.ito.tu-darmstadt.de/publs/papers/sac2002.pdf (16 May 2002).

Tidwell, T. Larson, R., Fitch K. and Hale, J.. "Modeling Internet Attacks". ISBN 0-7803-9814-9. June 2001. URL:

http://www.itoc.usma.edu/Workshop/2001/Authors/Submitted_Abstracts/paperT1C1(50).pdf (16 May 2002).

Walker, John. “Security Event Correlation: Where are we now”. 2001. URL: http://download-src.netiq.com/Library/white_papers/Security_Event_Correlation-Where_Are_We_Now.pdf (16 May 2002).

Last Updated: May 27th, 2017

Upcoming Training

SANS Stockholm 2017	Stockholm, Sweden	May 29, 2017 - Jun 03, 2017	Live Event
SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS Charlotte 2017	Charlotte, NC	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Rocky Mountain 2017 - SEC401: Security Essentials Bootcamp Style	Denver, CO	Jun 12, 2017 - Jun 17, 2017	vLive
SANS Secure Europe 2017	Amsterdam, Netherlands	Jun 12, 2017 - Jun 20, 2017	Live Event
Community SANS Portland SEC401	Portland, OR	Jun 12, 2017 - Jun 17, 2017	Community SANS
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Minneapolis 2017	Minneapolis, MN	Jun 19, 2017 - Jun 24, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, Australia	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Paris 2017	Paris, France	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, Japan	Jul 05, 2017 - Jul 15, 2017	Live Event
SANS Los Angeles - Long Beach 2017	Long Beach, CA	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Phoenix SEC401	Phoenix, AZ	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Munich Summer 2017	Munich, Germany	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, Singapore	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Minneapolis SEC401	Minneapolis, MN	Jul 10, 2017 - Jul 15, 2017	Community SANS
Mentor Session - SEC401	Macon, GA	Jul 12, 2017 - Aug 23, 2017	Mentor
Mentor Session - SEC401	Ventura, CA	Jul 12, 2017 - Sep 13, 2017	Mentor
Community SANS Atlanta SEC401	Atlanta, GA	Jul 17, 2017 - Jul 22, 2017	Community SANS
Community SANS Colorado Springs SEC401	Colorado Springs, CO	Jul 17, 2017 - Jul 22, 2017	Community SANS
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Jul 24, 2017 - Jul 29, 2017	Community SANS
SANSFIRE 2017 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Jul 24, 2017 - Jul 29, 2017	vLive
Community SANS Fort Lauderdale SEC401	Fort Lauderdale, FL	Jul 31, 2017 - Aug 05, 2017	Community SANS
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event