



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.

Interested in learning more?

Check out the list of upcoming events offering
"Network Monitoring and Threat Detection In-Depth (Security 503)"
at <http://www.giac.org/registration/gcia>

SIEM Based Intrusion Detection with Q1Labs Qradar

GIAC GCIA Gold Certification

Author: Jim Beechey, beechey@northwood.edu
Advisor: Rick Wanner

Accepted: February 12, 2010

Abstract

Attackers continue to become more skilled in their ability to penetrate organization's networks. Defenders need intelligent systems which provide meaningful data to detect advanced attacks. SIEM solutions are great tools for any security team. However, getting the most out of a SIEM solution requires focus on reporting, correlating and analyzing events across security systems. This is especially important when looking at intrusion detection. Today's attacks routinely bypass signature based systems and, therefore, require additional data sources beyond simply detecting specific attack traffic. Spending the time and effort to fully develop the correlation and reporting aspects of a SIEM can dramatically improve a team's ability to detect a compromise. While this paper focuses on Q1Labs Qradar, the intent is to provide rules and alerts which could also be used in other environments.

1. Introduction

There is no question that preventing attacks is the preferred outcome for security practitioners. The problem is that no matter how much time and money are spent on prevention technologies, eventually, prevention will fail. “This principle doesn’t mean you should abandon your prevention efforts. As a necessary ingredient of the security process, it is always preferable to prevent intrusions than to recover from them.

Unfortunately, no security professional maintains a 1.000 batting average against intruders. Prevention is a necessary but not sufficient component of security.” (Bejtlich, 2004)

Unfortunately, detecting successful attacks is increasingly difficult due to the level of sophistication and targeted nature employed in today’s attacks.

Gone are the days of dealing with simple defacements and script kiddies. Today’s attackers are highly organized and can be well funded. Attacks have evolved from simply being an annoyance to having the potential for significant financial impact to a business and even reaching the level of national security concern. Mandiant, a security consulting firm for Fortune 500 Corporations and the US Government, categorizes intrusions into three different levels, each having a different purpose and level of sophistication. See Figure 1 for details (Harms, 2008).

These increasingly complex attacks require much more than signature based solutions can provide. Individual security solutions, such as antivirus, are often easily bypassed via various techniques. According to Graham Ingram, General Manager of the Australian CERT, “The most popular brands of antivirus on the market... have an 80 percent miss rate. That is not a detection rate that is a miss rate”. (Kotadia, 2006) While

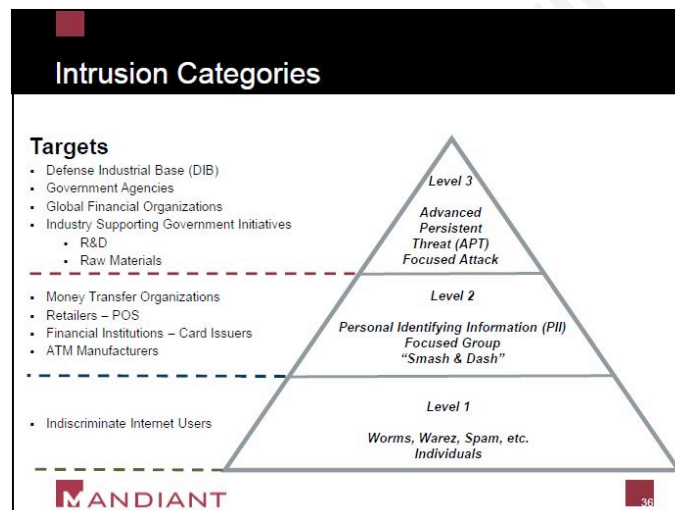


Figure 1 (Harms, 2008)

IDS does employ other techniques beyond signature detection alone, their success rate can also be limited, especially after the initial compromise. Attackers often use valid credentials to move around networks and transfer data using normal methods. Both cases would be very unlikely for IDS to catch. Therefore, organizations need systems which provide an overall view of the entire network. Richard Bejtlich, in his book The Tao of Network Security Monitoring says “defensible networks can be watched. A corollary of this principle is that defensible networks can be audited. ‘Accountants’ can make records of the ‘transactions’ occurring across and through the enterprise. Analysts can scrutinize these records for signs of misuse and intrusion.” (Bejtlich, 2004) Organizations need to take input from various security solutions and correlate events in order to detect potential compromises. Security Information and Event Management (SIEM) systems provide this capability and should be leveraged in order to maximize efforts in detecting today’s advanced threats.

While SIEM solutions offer many benefits to the overall security of an organization, they are often not funded or prioritized most heavily based upon security. A compliance requirement most often drives the purchase and implementation of these systems. “The primary driver of the North American SIEM market continues to be regulatory compliance. More than 80% of SIEM deployment projects are funded to close a compliance gap.” (Nicolett, 2009) At first glance this may not seem to be a problem. After all, funding a security project can be a major challenge given their cost to an organization. However, SIEM projects funded by compliance will tend to be focused on compliance, potentially at the expense of security. Marking the proverbial compliance check box and moving on to other issues could be a costly mistake. Organizations with compliance requirements should ensure that the project also impacts security operations and incident response before considering a SIEM project successful.

SIEM solutions, including Q1Labs Qradar, typically offer both reporting and alerting capabilities. Organizations should use both in detecting security incidents, however the decision about when to use one over the other is a decision best made by each individual organization. For instance, a large company with a 24 hour security operations center would likely want to employ more alerting capabilities in order to limit the time between compromise and detection. However, a smaller organization with a

single information security staff member may decide that receiving daily reports covering questionable activity may be the most effect method for prioritizing security activities. Regardless of the method of notification, the techniques discussed apply to both scenarios and should work from either an alerting or reporting perspective. Additionally, while the Q1Labs Qradar SIEM is our focus, these techniques should be effective for just about any centralized logging infrastructure and even systems with logs in multiple places. The only requirement is to be able to query data across multiple systems and data stores.

Another pitfall SIEM implementations face is not taking into account the post-implementation human resource requirements. SIEM vendors tout various alerting capabilities and correlation engines, however no implementation can be successful without being tuned and tailored to the needs of the organization. This is especially true when tuning the system to detect incidents. Effective detection requires knowledge of the existing infrastructure within the organization. After presenting a previous paper on SIEM implementations, I was approached by numerous people who ultimately had the same issue. “We’ve implemented a SIEM, now what?” People seemed to be inundated with alerts and frustrated with the lack of reliable and actionable data. Regardless of the system being implemented, organizations should be able to create custom reports and alerts to detect attacks accurately and efficiently.

The following examples come from real world experiences managing a multi-campus university network. In order to provide additional context, sections include real world examples using Q1Labs Qradar to detect an intrusion based upon the techniques discussed. The security challenges on a university network are very interesting as various constituents have different security requirements. While there are university owned systems on which we can impose security controls similar to our corporate counterparts; we also have residence halls which we must provide network access to non-university owned computers. The challenge is detecting and responding to compromise accurately and efficiently. Qradar is the primary resource for accomplishing this task.

2. System Setup

SIEM solutions can certainly mean different things to different people. For the purposes of our discussion, a SIEM will be a system capable of receiving logs from virtually any device, operating system or application in the enterprise. Most people think of network security related items such as firewall, intrusion detection and VPN logs first when considering SIEM. This is perfectly fine, but organizations need not limit themselves to these technologies. The goal should be to have each and every log in the enterprise collected in the SIEM.

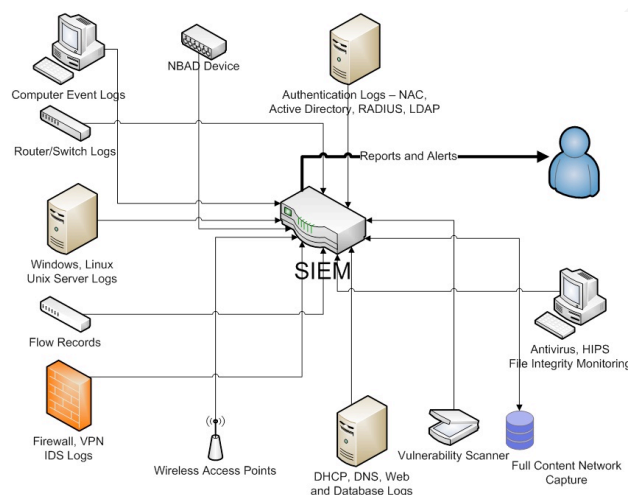


Figure 2

Devices and applications which do not necessarily have a focus on security still can add significant value during an investigation. In addition to log collection, ideally a solution will include the collection of session data and access to full content network captures.

SIEM solutions do not require session data, also called flows; however the ability to access this information can dramatically improve the capability of the system. “The basic elements of session data include the following: Source IP, Source port, Destination IP, Destination port, Protocol (e.g., TCP, UDP, ICMP), Timestamp, generally when the session began and measure of the amount of information exchanged during the session.” (Bejtlich, 2006) The most common flow records are Cisco’s Netflow, however there are several other options including sFlow and Jflow. These technologies collect traditional session data without any application data. There are also more specialized NBAD products such as Q1Labs Qflow collectors which allow for capture of a portion of the application data within the flow record. This can greatly assist in determining whether an anomaly is an incident or false positive.

Full content network captures are typically not built directly into SIEM products. However, SIEM solutions often have capabilities to integrate with systems providing full content captures. For example, NetWitness NextGen products provide full content network captures for network forensics purposes. NetWitness has an application called SIEMLink designed to integrate with an organization's SIEM. "SIEMLink is a light-weight Windows application designed to act as a transparent, real-time translator of critical security event data between Web-based consoles, such as security event and information management (SIEM) systems and network and system management (NSM) programs." ("Netwitness total network," 2009)

Collecting full content network captures is certainly not a simple task due to issues of performance, storage and politics. However, if organizations can address the issues involved, full content network captures will provide significant benefits in explaining what really happened during a potential incident. "By keeping a record of the maximum amount of network activity allowed by policy and collection hardware, analysts buy themselves the greatest likelihood of understanding the extent of intrusions." (Bejtlich, 2004) For the purposes of this discussion we will assume the SIEM has access to log and flow data, leaving full content network captures as an optional method for further determining the extent of an intrusion.

3. Suspicious Traffic and Services

Developing various reports and alarms for intrusion detection can seem like an overwhelming task. The best approach is to start with some simple alerts and build more advanced correlations from there. These reports are very simple to create, yet still can be highly effective in locating compromised machines. While many organizations are likely blocking much of the traffic discussed, reporting on its attempted usage is still valuable to detect successful attacks. In fact, one could argue that traffic which is prohibited by policy or technical controls, yet still exists on the network, may be more indicative of malicious activity.

3.1. SMTP, IRC and DNS

A great place to start is outbound SMTP traffic. “Keep an eye out for a massive amount of SMTP outbound traffic. Such patterns, especially coming from machines that are not supposed to be SMTP servers, will likely point to a malware spam bot that has implanted itself in your organization.” (“Shadowserver foundation information,” 2009) Monitoring outbound email traffic, regardless of whether the traffic is allowed or blocked by the firewall, is a highly effective method for detecting compromised hosts. This can be done by monitoring firewall or flow logs. Create a report or rule to monitor any outbound traffic destined for port 25. However, be sure to exclude valid SMTP senders such as mail servers, web servers which email forms and vulnerability scanners. Daily reports covering the previous 24 hours are effective or rules can be created to flag an alert after a certain threshold has been crossed. I’ve used daily SMTP reports for years in a university dorm network with very high success rate. Standard practice for our team is to assume any machine generating 250 or more SMTP events in a 24 hour period is compromised. Most often, the numbers will be much higher, likely in the thousands of events.

Internet Relay Chat is a protocol used to chat via the Internet, most often by technically oriented people. IRC also is “one of the very first types of botnet: bots were controlled via IRC (Internet Relay Chat) channels. Each infected computer connected to the IRC server indicated in the body of the bot program, and waited for commands from its master on a certain channel.” (Kamluk, 2008) IRC uses a range of ports, but most often port 6667. The existence of IRC traffic alone is not a guarantee of malicious activity as IRC is still used for legitimate communications. An effective method for determining what traffic is malicious is simply asking the user of the computer if they know what IRC is or if they are using it. Regardless, monitoring outbound traffic to ports 6660-6669 from firewall logs is still a good idea to detect potentially compromised machines.

DNS activity is most certainly not malicious by itself. However, only DNS servers should be communicating externally via DNS. Client workstations or non-DNS servers should not, and therefore, may be a sign of compromised machines. Specific Trojans, called DNS changers, are designed to change a host computer’s DNS server

settings so clients resolve domains from external servers and can be redirected to malicious sites. “Check the machine's default DNS resolution servers. Are they what you would expect to see (a company's or ISP's DNS servers, or that of your internal LAN's router?) If not, malware may be redirecting DNS requests to a shady source.”

("Shadowserver foundation information," 2009) Instead of checking individual machines, an organization can use SIEM to monitor their entire organization. Create a SIEM rule or daily report to monitor outbound traffic to port 53, excluding DNS servers. Any systems which show up on the report should be investigated further for potential compromise.

These three methods for detecting compromised machines are certainly very basic, but can still be effective and should be the start of any log based intrusion detection planning. A simple report, generated every 24 hours, with these three criteria is a good starting place. The following report is an example of a daily user defined report in Qradar detailing external SMTP activity (destination port 25) by source IP address. Addresses for known SMTP servers are excluded. This report identified three infected clients located in our student dormitories; source IP addresses have been removed.

Outbound SMTP by Source IP Address
Generated: Jan 14, 2010 1:01:23 AM

Outbound SMTP
SMTP Senders
Jan 13, 2010 12:00:00 AM - Jan 14, 2010 12:00:00 AM

Source IP	Event Name (Unique Count)	Device (Unique Count)	Event Count (Sum)	Start Time (Minimum)	Category (Unique Count)	Source Port (Unique Count)	Destination IP (Unique Count)	Destination Port (Unique Count)	Username (Unique Count)	Magnitude (Minimum)	Count
Multiple (1)	Multiple (1)	Multiple (1)	521 632	2010-01-13 03:23:42	Multiple (1)	Multiple (64 496)	Multiple (52 195)	Multiple (1)	Multiple (1)	4	474 744
Multiple (1)	Multiple (13)	Multiple (13)	50 362	2010-01-13 04:10:26	Multiple (1)	Multiple (25 443)	Multiple (94)	Multiple (1)	Multiple (1)	5	25 684
Multiple (1)	Multiple (1)	Multiple (1)	944	2010-01-13 22:41:22	Multiple (1)	Multiple (869)	Multiple (613)	Multiple (1)	Multiple (1)	4	884

Figure 3

3.2. Suspicious Outbound Internet Traffic

In addition to looking for potentially malicious traffic across the network, incident detection plans should also look for general traffic which does not fit the profile of the originating machine. For instance, there are certain systems and devices that really should not be making outbound internet connections. For example, printers likely should not require outbound internet access and can be used for malicious purposes. “Network printers have long been used as jump off points in exploiting networks and for storage of hacking tools and data.” (Danford, 2009) Simple alerts can be created to notify the

Jim Beechev. bechev@northwood.edu

analyst of outbound connections from these devices, assuming they are properly defined and segregated within the enterprise.

Embedded devices also pose a potential risk and likely do not require extensive Internet access. Unfortunately, these devices are also often lacking the patch management procedures of their PC and server counterparts. The Conficker worm garnered much attention during 2009 and for good reason. Our network went relatively unscathed to the worm due to timely patching. The one area we did see compromised machines were several Windows embedded devices used for controlling classroom technology equipment. In fact, we believe the initial infection point was a device which was shipped to us pre-loaded with Conficker. These compromises were detected because the embedded system began attempting connections outside of their permitted network segment.

Organizations which segment their devices properly can use Qradar to monitor these network segments for suspicious activity. This could be both external Internet access or even simply network connections into or out of the local subnet. Some organizations will filter these subnets with firewalls or router ACLs. In this case, Qradar has a built in rule for watching denied connections called “Recon: Excessive Firewall Denies from Local Host”. This rule will fire if a single source IP creates 40 denied connections to at least 40 destinations in five minutes. In order to create a rule for specific segments, the rule can be copied, then modified to include specific source IP ranges and different detection settings if so desired.

Servers are another area which you can use system profiling of network activity to detect compromise. Do the vast majority of your servers really need to make outbound Internet connections? For those that do required Internet access, to how many addresses or domains? This report will take a bit more time to put together as you’ll likely need to filter out a few updates sites, outgoing SMTP for mail relays, etc. This is an area I would recommend getting system administrators involved if possible. Do some initial tuning of the obvious false positives, then ask system administrators to review a quick daily report of the systems for which they are responsible and alert the security team to any anomalies. If possible, automate the process to create the report and create a regular call in the IT ticketing system. Certainly, some occasional audits to make sure the process is

being followed will be required. However, getting non-security focused staff involved in the process only helps build more awareness of security related issues.

3.3. New Hosts and Services

The favorite saying of one of my colleagues is “What’s changed”. His question anytime something isn’t working correctly. This analogy very much applies to intrusion detection. New services can be indicative of recently installed backdoors or accidentally installed services which could become targets for new attacks. New hosts on the network could be a rouge device like a wireless access point or a non-standard workstation. While these events are not a guarantee of compromise; removing non-sanctioned devices can help prevent future attacks.

Qradar can collect this data from a variety of sources including integration with port scanners such as Nmap, or vulnerability scanners like Nessus or Qualys and the collection of “passive” data based upon flows. The simplest and one of the most effective methods is integration with an already existing Nmap scanner. This integration can be accomplished by defining your Nmap scanner using the VA Scanner button on the Admin tab of Qradar. Once the scanner has been setup with the proper credentials, scans can be scheduled from with the VA Scan section of the Asset Tab. Qradar will log onto the defined Nmap scanner, launch the scan, retrieve the results and publish the data within the appropriate asset record. Figure 4 shows a sample asset record after being scanned with Nmap.

Port	Service	OSVDB ID	Name	Description	Risk / Severity	Last Seen	First Seen
135	mrsrpc				1	2010-01-29 00:06:05 (Active)	2010-01-29 00:06:05 (Active)
139	netbios-ssn				1	2010-01-29 00:06:05 (Active)	2010-01-29 00:06:05 (Active)
205	ssl				1	2010-01-29 00:06:05 (Active)	2010-01-29 00:06:05 (Active)
445	netbios-ssn				1	2010-01-29 00:06:05 (Active)	2010-01-29 00:06:05 (Active)
2869	http				1	2010-01-29 00:06:05 (Active)	2010-01-29 00:06:05 (Active)
3389	ms-term-serv				1	2010-01-29 00:06:05 (Active)	2010-01-29 00:06:05 (Active)

Figure 4

Qradar includes a standard rule to detect the presences of a new service in the DMZ. The “DMZ” is a network object which is defined during the tuning phase of a Qradar deployment. However, this rule can be easily modified or duplicated to watch for changes to any network segment or specific hosts. Figure 5 shows an offense generated when Qradar detected a new service in the DMZ based upon regular port scans using the Nmap integration. Double clicking the “new port discovered” event (see arrow) would provide details regarding the port discovered which in this can was port 3389 (Microsoft Remote Desktop).

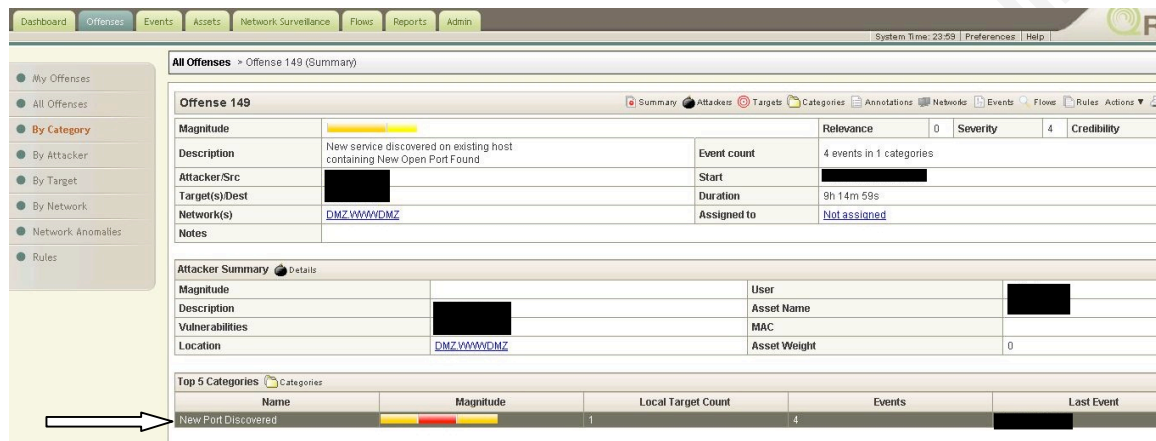


Figure 5

The detection of new devices using Qradar uses the same Nmap scan data and another pre-defined rule. From a security perspective, this rule can be used to find rogue devices such as a rogue access point. In addition to finding rogue devices these scans are highly effective in finding devices with improperly configured network settings. Many organizations subnet various devices to provide additional separation and security controls. Devices such as printers, embedded systems, HVAC, etc often have their own network segments. In the university environment keeping student owned computers on the proper VLANs and segregated from faculty and staff networks is important. In addition to the canned rule Qradar provides, organizations can increase the effectiveness of their scans by looking for specific devices or the existence of devices which do not match what should exist in the subnet. Qradar can detect and exclude an operating system by adding the following criteria to any rule. From within the “Rule Test Stack Editor” in Qradar, select the Test Group “Event Property Tests” and select the criteria “when the username matches the following regex”. Next, change “username” to “OS”

and change “regex” to a regular expression appropriate for the operating system you would like to detect or exclude. However, I prefer to determine operating system rules based upon open/close ports on the asset as this method has proven more effective during our testing.

For instance, an organization with subnets dedicated to VOIP handsets could create a custom rule similar to Figure 6. In this case, VOIP phones used do not have any listening ports; therefore the rule detects the presence of a device in the VOIP subnet with an open port. If such a device is detected, an offense is generated for investigation. Conversely, the custom rule in Figure 7 is designed to catch any non-windows device inside of a subnet designated for windows-based computers. This helps to detect rogue access points, printers in the wrong subnet and non-windows personal devices.

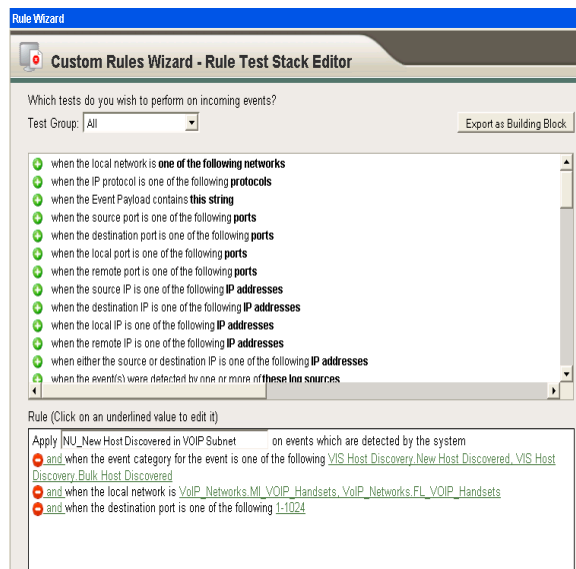


Figure 6

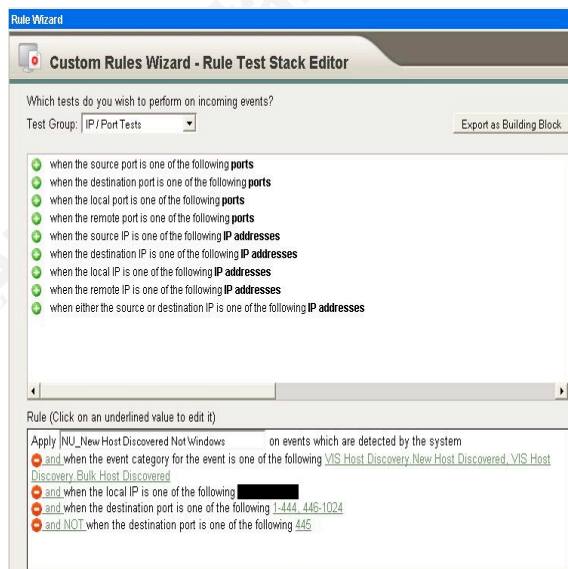


Figure 7

3.4. Darknets

Darknets are a classic method for detecting suspicious traffic. The concept is quite simple. Create network segments inside your infrastructure which are routable, however have no systems or devices setup to use the local network. Therefore, no system on your network should be attempting to access anything within the Darknet. “Any packet that enters a Darknet is by its presence aberrant. No legitimate packets should be sent to a Darknet. Such packets may have arrived by mistake or misconfiguration, but the majority of such packets are sent by malware. This malware, actively scanning for

vulnerable devices, will send packets into the Darknet, and this is exactly what we want.” (“The Darknet project,”) Qradar can monitor traffic events and flow records to watch for systems attempting to access predefined Darknet addresses. During the initial setup of your network hierarchy, Darknets can be defined. Qradar will then monitor these network segments based upon the rule “Suspicious Activity: Communication with Known Watched Networks” and generate an offense accordingly. While the default rule will work, it includes both watched network lists and Darknet addresses in the same rule. I prefer to have these separate and therefore create a customized rule for these categories.

The example in Figure 8 is not only an example of the Darknet address rule firing, but also a good example of how Qradar correlates various suspicious network activities across both events (firewalls in this case) and flow data. In addition, due to integration with identity information the username of the person currently logged into the internal, attacking host is also displayed.

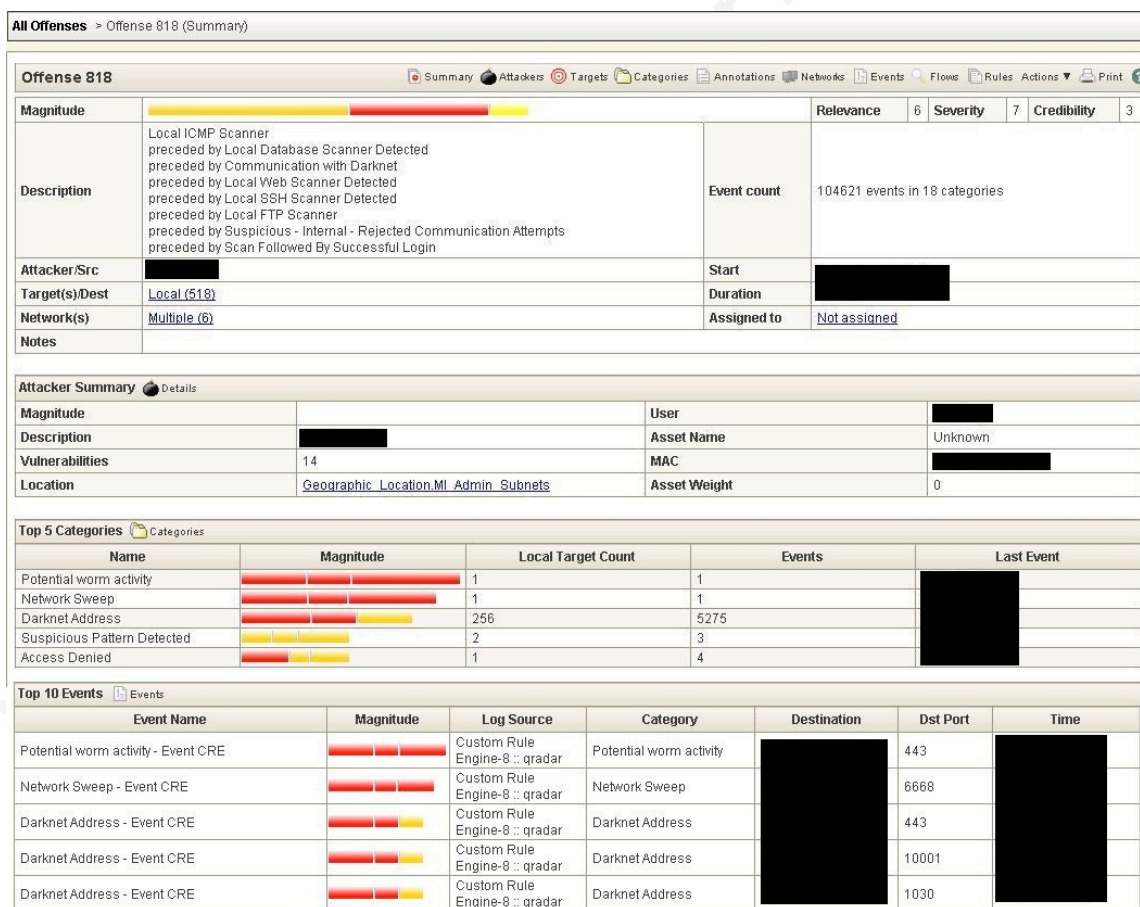


Figure 8

4. Authentication, Accounts and Remote Access

While attacks continue to evolve in their complexity, compromised accounts continue to be effective method for intruders. Compromised accounts may not always be the initial attack vector, however they are often used to move throughout the organization or elevate the privileges of the attacker. Therefore, proper auditing of authentication attempts, account changes and access tracking can be highly effective in detecting intrusions. Please note that each time a Windows event ID is discussed there will be two numerical entries. Three digit entries correspond to Windows XP or Server 2003. Four digit entries apply to Windows Vista, 7 and Server 2008.

4.1. Brute-force Attacks

According to the SANS Institute Top 20 list, “Brute-force attacks against remote services such as SSH, FTP, and Telnet are still the most common form of attack to compromise servers facing the Internet.” (“SANS: Top Twenty,” 2007) SIEM is a perfect place to collect failed authentication attempts which could be indicative of a brute force attack. Q1Labs Qradar will take authentication events from a variety of sources such as SSH, FTP, Linux and Windows, and categorize events together so reports and alarms are easily developed. This categorization process is known as normalization.

A good first step in identifying brute-force attacks is to understand what is typical for your organization. Gathering statistics regarding daily failed login attempts by device is a great statistic to have. Once you understand what’s normal for your organization, identifying attacks can be much easier. Creating this report is straightforward inside of Qradar. Create an event search covering the past 24 hours using the following criteria: Category = Authentication (High Level) and “Failed Authentication” (Low Level). Under the “Column Definition” section of the event search, have the data sorted by device based upon the sum of the event count from high to low. Another helpful variation of the above report would be to sort the data based upon source IP address rather than device.

Organizations who lock accounts for a period of time after a certain number of failed authentication attempts may also find daily statistics on locked accounts useful. Organizations with Active Directory can accomplish this by reporting on Windows event

ID 644/4740 or, within Qradar, create an event search with log sources of Active Directory controllers and an event name of “User Account Locked Out”. Save the event search and create a daily report based upon the search. This data, collected over a period of time, should provide enough data to estimate how many accounts generally are locked within a 24 hour period. Use this baseline to compare reports and look for problems or create an offense watching for more than X number of locked accounts within a given timeframe. These numbers will be unique to each organization, however setting this kind of an alert will help with early detection of brute force attacks.

Beyond daily reports, Qradar provides several out of the box rules aimed at identifying brute-force password attacks. The general purpose of these rules is to detect a certain number failed login attempts followed by a successful login. These rules are highly effective for systems without significant login activity such as routers and firewalls. However, detection is much more difficult on high activity systems such as a web-based email server. Separating a brute-force attack from a user who forgot to change their Blackberry password is challenging. The following offense shown in Figure 9 was generated by Qradar based upon a successful brute-force attack against a Cisco device using SSH. The rule which fired is using the canned logic for brute force attacks, however is customized to watch for attacks specifically aimed at network equipment and alerts are sent to both the network and security teams. Customizing the rules for specific device helps separate alerts to help in prioritization.

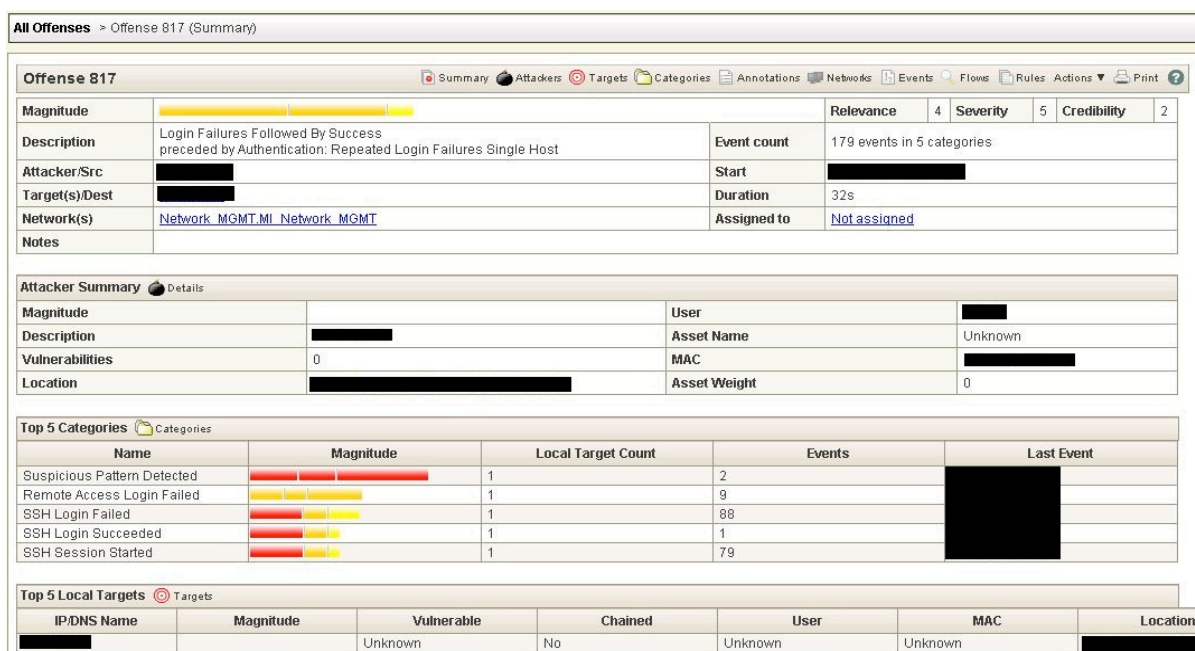


Figure 9

Detecting brute-force attacks against general user accounts is certainly important, however detecting attacks against privileged accounts is critical. A good starting point for detecting these attacks is to create a list of accounts throughout the organization with elevated privileges. The list should include obvious accounts such as root and administrator, but also system level accounts used for services and database access. Windows shops will want to include any account with domain/enterprise admin access as well. In fact, one could argue that most IT staff will have some kind of elevated privileges worth monitoring. Custom brute-force rules can then be developed to look for attacks on privileged accounts. Copy the built-in Qradar brute-force rules and add additional requirements such as a list of critical usernames and/or system logs. You may also consider tuning the brute force parameters for number of attempts in a given timeframe to be more sensitive given the accounts being watched.

4.2. Windows Account Creation and Permissions

Detecting brute-force attacks and various authentication events is one method for detecting a compromised account. Another method for detecting potential compromise is tracking account creation and privilege changes within an Active Directory domain. These events can help detect when an attacker is attempting to increase their privileges or

access sensitive resources. However, monitoring account activity within an Active Directory domain really requires logging beyond just Active Directory controllers. Collecting Windows event logs from all systems, including individual workstations, is the best way to get a full picture of what is occurring on an organizations windows network. The following examples assume collection from all sources within the domain.

Creating accounts, while not necessarily the initial intrusion vector, can be a key method for maintaining access. However, creating an account is also one of the most basic functions within IT. Separating the malicious from the mundane can be difficult. The most basic method would be to create a daily report listing the account creations and what account was used to create the account. This report can be automatically created using Qradar's built in event search and emailed to system administrators for review. If your organization has a strict naming convention, such as six characters followed by two digits, a rule could be created to flag account creations which do not meet organizational criteria. Since many organizations have automated their employee account creations utilizing nightly scripts or automatic triggers from another system, rules could be created to list accounts created outside the nightly script timeframe or created outside of the automated process. Security teams need to understand their organization's account creation process and build rules based upon their specific requirements in order to best detect rogue account creations.

Most organizations are likely to prohibit or discourage the creation of local accounts. Organizations those that do, should consider creating an offense for the creation of a local account. This rule will help detect policy violations, but also attacks in which a local account is created in order to maintain or elevate access.

Next, let's take a look at changes to an account's privileges focusing on Windows environments and Active Directory. Windows will log various events of interest in detecting attempts to change access permissions for an account. Qradar can be used to create a daily report of these event IDs for review or create an offense for immediate review. There are going to be numerous events which are valid during normal operations. The key is identifying important areas to focus on. Most Active Directory domain permissions come from group membership, therefore monitoring changes to key groups is important in detecting intrusions. A good starting place is to create a rule inside

of Qradar to alert when changes to key groups are made. The following custom rule in Figure 10 watches logs for a member being to a global group and then checks for “Domain Admins” within the payload of the event. Thus, whenever an account is added to the Domain Admins group, an offense is generated inside of Qradar and the appropriate staff notified via email. Organizations can add other key security groups to this rule for further coverage. Additional rules can also be created for local groups to cover items such as local administrator access to PCs.

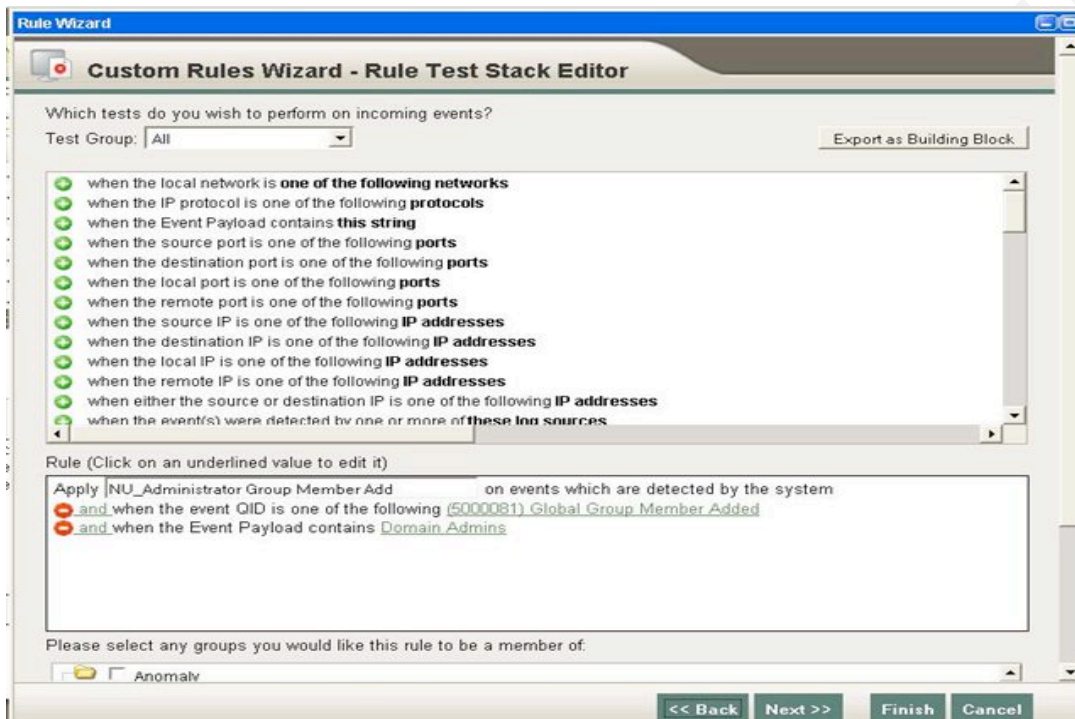


Figure 10

4.3. Foreign Country Logins

Another method for detecting intrusions using valid credentials is to use SIEM to correlate logins with geo-location data. “One of the use cases we tackled was the monitoring of login attempts from foreign countries. We wanted to keep a particularly close watch on successful logins from countries in which we don't normally have employees in... we had to have the ability to extract usernames and IP addresses from these logs; and, we had to have the ability to map an IP address to a country code.” (Bejtlich, 2008) Qradar is capable for providing similar data. First, within the rules section of the offense tab, edit the building block titled “Category Definition: Countries

with no Remote Access” and enter countries where logins should not come from. Next, enable the rule “Anomaly: Remote Access from Foreign Country” and any login events from banned countries will become offenses. If needed, you can also further customize the alert to specific log sources. For instance, in our University we would not be able to track international logins for our web-based email given the large contingent of international programs and students. However, monitoring our VPN and limiting access to only those areas with current operations provides valuable data. Certainly, the value of this capability depends upon the organization in question. However, organizations which are predominantly domestic or do business in a limited number of countries may find the services helpful. Also, organizations may be able to target logins from specific countries known to be hotbeds for malicious activity.

5. Adding Context and Correlation to IDS Alerts

IDS and IPS solutions, whether open source or commercial, can create a mountain of alerts to classify and respond to. “On any given network, on any given day, Snort can fire thousands of alerts. Your task as an intrusion analyst is to sift through the data, extract events of interest, and separate the false positives from the actual attacks.” (Beale, Baker, et al, 2006) SIEM can help greatly in dealing with IDS alerts as they typically offer a variety of options for reporting on and analyzing IDS alerts. This can be done in a variety of ways including different methods of reporting, adding knowledge of the target operating system or applications and including data regarding vulnerabilities which exist on the target system.

First, consider the benefit of customized reporting a SIEM can provide. IDS can be a noisy technology and focusing effort on the most critical alerts can help find events worth investigating. For example, Qradar can be used to filter IDP events and create a report for high value systems within the organization. Qradar allows users to assign an “Asset Weight” to each asset inside of the asset profile shown in Figure 4. These asset weights can then be used in various event searches and reports. For instance, an offense or alert could be created anytime an exploit is seen against a high value target. This can help prioritize security analyst’s time appropriately. Commercial IDS systems likely have solutions capable of doing this work for you; however why not do so in a

centralized console where the rest of the organizations logs and alerts reside. Also, due to support for multiple IDS solutions, Qradar can allow central collection and correlation across multiple kinds of sensors.

Second, look for ways to add knowledge of the target operating system or installed applications in order to make the report more effective. Qradar will parse supported IDS logs and then categorize events based upon data provided. This can allow users to filter out events which are not valid for their organization. For instance, one could create a report filtering out “Unix” alerts for your subnets containing Windows servers or email a report of web applications attacks detected to the web development team.

Third, further data may be gathered via integration with vulnerability scanners. “Keep a list of vulnerable systems and refer to it when attacks occur. If you know your host is not vulnerable to a particular attack, you can rest assured that the attack was not successful.” (“The Truth about,” 2001) Assets must first be scanned with a supported vulnerability scanner from within Qradar. Next, Qradar watches intrusion detection logs for exploits targeting systems which are vulnerable to the attack. Qradar has several canned rules to help provide this capability. The system includes rules for “Target Vulnerable to Detected Exploit, Target Vulnerable to Detected Exploit on Different Port and Target Vulnerable to Different Exploit than Detected on Attacked Port”. The rule “Target Vulnerable to Detected Exploit” obviously has strong value; however don’t discount the other rules. IDS systems may not always be able to detect the exact exploit an attacker is delivering, but may still detect malicious activity relating to the attack such as the existence of shellcode or protocol anomalies.

6. Web Application Attacks

Web application attacks are a key vector for compromise in today’s enterprise networks. Attacks such as SQL injection allow attackers to take control of internal resources via vulnerable public facing web applications. Once the attacker has access to the internal database server, he can attack other internal resources or exfiltrate sensitive data. The Verizon Business 2009 Data Breach Investigations Supplemental Report states that SQL injection was a “factor in 18% of breaches in caseload” and a “factor in 79% of

records compromised in caseload”. (Baker, Hylander, & Valentine, 2009) Cross-site scripting attacks allow attackers to compromise client systems visiting trusted resources. Compromise could occur on internal corporate systems or customers visiting your organizations web site.

Certainly the most effective method for dealing with web applications attacks is proper development practices. The goal should be to eliminate these vulnerabilities. However, if vulnerabilities do exist, whether known or unknown, how do we detect attacks against them? Web application attacks are very challenging as they use the same ports and services to conduct malicious activity as are used for non-malicious activity. From a logging perspective, there are several options for monitoring logs for malicious activity. Logs can of course be collected from the web server itself and is certainly the most common location. However, web server logs have one major disadvantage. “Web server logs do not contain any data sent in the HTTP header, like POST parameters. The HTTP header can contain valuable data, as most forms and their parameters are submitted by POST requests. This comes as a big deficiency for web server log files.” (Meyer, 2008) Another, more effective, location for generating valuable web application log files is a web application firewall. “WAF log files contain as much information as those from a web server plus the policy decisions of the filter rules (e.g. HTTP request blocked; file transfer size limit reached, etc.). A WAF provides a wealth of information for filtering and detection purposes and is thus a good place for the detection of attacks.” (Meyer, 2008) Organizations with a WAF in place, which is supported by their SIEM, should consider doing their log analysis on those log files. However, a WAF does require additional investment and is not an option for all organizations. Regardless, organizations can analyze whatever logs are available to detect many common web attacks.

6.1. SQL Injection and Cross Site Scripting

Detecting SQL injection and cross site scripting attacks via web server logs can be challenging due to the propensity for false positives and ability for attackers to encode attacks. Therefore, some knowledge of the organization’s applications will be helpful in tuning detection methods. Detection of web application attacks will focus on patterns known to be SQL injection or cross site scripting attacks. Qradar allows for searching the

payload of log files based upon regular expressions. Therefore, the analyst can create log searches or alerts looking for specific attacks. The following regular expressions were published on securityfocus.com. (Mookhey, 2004)

SQL Injection

- `/(\%27)|(\')|(\-\-)|(\%23)|(\#)/ix`

This regular expression will detect the comment characters, single quote (MS-SQL) and double-dash (Oracle) and their hexadecimal equivalents. These characters are used to terminate queries and often part of SQL injection attacks.

- `/((\%3D)|(\=))[\^n]*(\%27)|(\')|(\-\-)|(\%3B)|(;))/ix`

“This signature first looks out for the = sign or its hex equivalent (%3D). It then allows for zero or more non-newline characters, and then it checks for the single-quote, the double-dash or the semi-colon.” (Mookhey, 2004)

- `\w*((\%27)|(\'))((\%6F)|o|(\%4F))((\%72)|r|(\%52))/ix`

“\w* - zero or more alphanumeric or underscore characters

(\%27)|' - the ubiquitous single-quote or its hex equivalent

(\%6F)|o|(\%4F))(\%72)|r|(\%52) - the word 'or' with various combinations of its upper and lower case hex equivalents.” (Mookhey, 2004)

- `/((\%27)|(\'))union/ix`

This will detect the single quote in ASCII or hex followed by the Union keyword. Other SQL commands can be substituted for union.

- `/exec(\s|+)(s|x)p\w+/ix`

This regular expression is specific to Microsoft SQL environments and will detect the EXEC keyword signifying that a Microsoft stored procedure is to be run.

Cross Site Scripting

- `/((\%3C)|<)((\%2F)|\/)*[a-z0-9\%]+((\%3E)|>)/ix`

This regex will detect simple XSS attacks looking for HTML opening and closing tags with text in between and their hex equivalents.

- `/((\%3C)|<)((\%69)|i|(\%49))((\%6D)|m|(\%4D))((\%67)|g|(\%47))[\^n]+((\%3E)|>)/I`

This regular expression will detect the “<img src” XSS attack

- `/((\%3C)|<)[\^n]+((\%3E)|>)/I`

“This signature simply looks for the opening HTML tag, and its hex equivalent, followed by one or more characters other than the newline, and then followed by the closing tag or its hex equivalent. This may end up giving a few false positives depending upon how your Web application and Web server are structured, but it is guaranteed to catch anything that even remotely resembles a cross-site scripting attack.” (Mookhey, 2004)

6.2. Web Application Honey Tokens

Beyond using regular expression to look for web application attacks, organizations have another option for detecting when someone is attempting to compromise one of their web applications. Web application honey tokens are intended to create data or portions of the web site which no normal activities should ever access. Therefore, if these fake items are accessed one can assume that an attacker is attempting some kind of malicious activity. The two ideas listed below come from a SANS Application Street Fighter Blog entry by Johanness Ullrich. (Ullrich, 2009)

First, create a fictitious “administration” web page and add the link to the disallowed section of the web server’s robots.txt file. After this is in place, check web server logs for anyone accessing the robots.txt file and later accessing the fake administration page. Second, add fake authentication credentials into the html source of a specific web page. Use the SIEM to query for anyone attempting to login using these credentials. Any IP addresses which attempt either of these two activities should be investigated further, added to your watch list and potentially blocked. Figure 11 shows an alert created to detect the robots.txt web honey token example. The rule also highlights the Qradar capability of creating multiple custom rules (red circles) and combining them into a function.

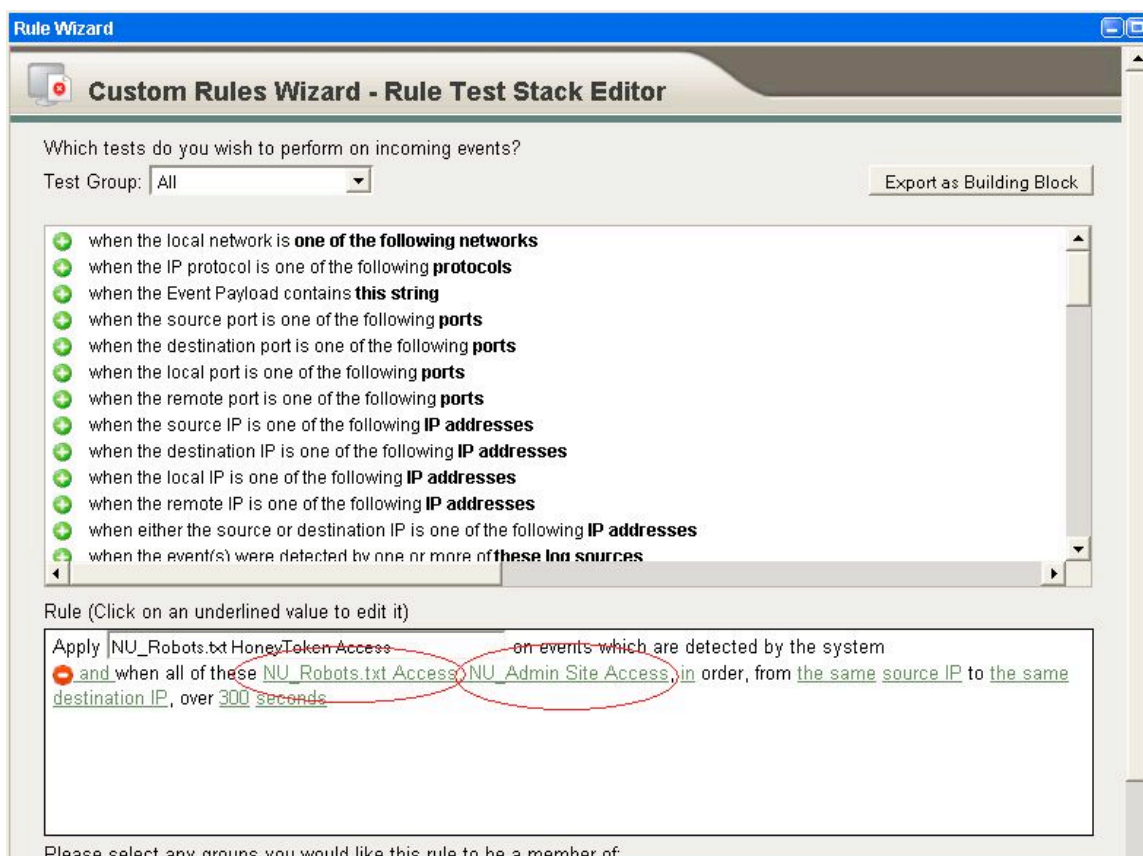


Figure 11

7. Data Exfiltration

Ideally, compromised systems would be detected and remediated quickly enough to limit any exposure of sensitive data. However, that is not always possible. Also, after a compromised is detected, analysts need to be able to determine if any data left the system in question or if the attacker used the compromised system to attack other targets. Security teams need to have systems in place to monitor network traffic effectively enough to address these challenges. The collection of session data at various locations throughout the network can be a tremendous help in achieving this goal.

The use of encryption in documents, archives and communications channels can make detection of sensitive data leaving the organization difficult with signatures and pattern matching alone. However, session data can still be used to determine in general terms what the attackers next steps were on the network. Large outbound transfers may point the investigation towards determining what data left the system. Conversely,

internal network traffic after exploitation may lead the incident response team to other targets. Therefore, one of the first steps after an incident has been declared is to collect as much session data as possible for all systems involved before, during and after the incident. Qradar provides a search capability for flow data in the same manner as standard log files can be searched. There are a multitude of options available including searching based upon IP address, ports, applications, number of bytes, flow direction, etc.

Session data is not only helpful after an incident has been identified, but can also be the reason for detecting an incident in the first place. For instance, why would a DNS server transfer a 50MB file outbound to a free file sharing service using SSL? Session data could highlight this anomalous activity assuming the correct reports or alerts have been developed. Intrusion detection based upon the amount of data transferred during a session will most often focus on outbound file transfers. Analysts will be looking for large outbound flows or flows with questionable destinations. The idea being, anyone intent on stealing corporate data must somehow transfer the data outside the network. If not done via physical means, then the Internet is the most likely option. Checking for session data is a great indicator of compromise because attackers cannot cover their actions via encryption. Qradar provides an alerting mechanism for network activity called sentries. Many sentries are created with the installation of a new system and custom sentries can also be developed.

Creating reports and alerts for intrusion detection based upon the size or destination of flows will take consistent tuning. Clearly, there are numerous possibilities for false positive. Also, the type of machine involved in the connection may also help in determining the likelihood of compromise. Again, the example used previously about a server making an outbound connection applies. Analysts will need to tune for false positives by identifying update sites, outsourcing relationships and commonly used services. Having some form of content data can be a significant help in quickly tuning false positives. Qradar includes a sentry to detect “External – Large Outbound Data Flow”. This sentry can be used to create an offense for such activity or add events to an existing offense to alert to possible data exfiltration after an incident.

Another technique for detecting intrusions based upon the size of network sessions is to look for large amounts of application data inside of protocols which should

have a limited size. Detecting these activities could be indicative of a covert channel. ICMP is an example of where this may apply. “Excessive amounts of data in ICMP traffic may indicate use of covert channels, especially if the content appears obscured. ICMP data that cannot be decoded is probably encrypted and encrypted content is a sure sign of a covert channel.” (Bejtlich, 2006) Sentries are available for several protocol related anomalies including: Unidirectional ICMP traffic

Frequency of requests can be another indicator of compromise. Security analysts need to understand what is “normal” for their network. Spikes in traffic or specific protocols should be a warning flag that something, possibly security related, is amiss on the network. These techniques can apply to the entire network, sub-networks or even specific hosts. “A covert channel may bear the headers and fields needed to look like DNS, but the content may be malicious. An internal workstation making very frequent DNS request may not be doing so for business purposes” (Bejtlich, 2006). Qradar offers many options configuring this kind of statistical data. Creating a statistical report of application usage is a good starting point. Once a baseline is developed, Qradar does have some capability to develop reports based upon deltas, or changes, in the data which could be very helpful in detecting anomalies.

Session duration is the final method for detecting intrusions from flow data. Protocols known for short session lengths could be analyzed for longer sessions in order to detect a possible covert channel. HTTP is a good example of a protocol that meets these criteria. “Web connections are usually short, even when HTTP 1.1 is used... A Web connection generally lasts several seconds. If an analyst notices a sustained outbound Web connection, something suspicious may be happening. An intruder may have crafted a custom covert channel tunneled over port 80.” (Bejtlich, 2006) Qradar comes with a predefined sentry called “Policy – External – Long Duration Flow Detected”. This rule will fire after a flow’s duration has exceeded 48 hours.

7.1. Tuning with content

A report detailing outbound flow data flagged a FTP connection where approximately 1.3 Gigabytes of data was transferred outbound from a University employee's computer late at night. The destination IP address did not resolve when queried via DNS. In addition, a WHOIS lookup on the IP address did not produce any relevant details. This situation would have been fairly labor intensive to resolve if not for the partial application data collected with the flow in question. Instead of having to further an investigation, we were able to quickly identify the transfer as non-malicious transfer.

The highlighted portion shows the file name of the upload. Since the file was uploaded from a basketball coach's computer, a few hours after a scheduled basketball game, we can be very confident that this exchange was not malicious. In addition, we can add the IP address to our whitelist of known good transfer IPs so future reports will not flag these events.

Flow Type:	Standard Flow	Protocol:	tcp_ip
Flow Direction:	L2R		
Source IP:		Destination IP:	
IPv6 Source:	0:0:0:0:0:0:0:0	IPv6 Destination:	0:0:0:0:0:0:0:0
Source Port:	3808	Destination Port:	21
Source Flags:	S,P,A	Destination Flags:	S,P,A
Flow Source:		Flow Interface:	eth1
Source QoS:	Best Effort	Destination QoS:	Best Effort
Source ASN:	0	Destination ASN:	0
Source If Index:	0	Destination If Index:	0
Start Time:	2009-12-09 23:16:48	Application:	FTP
End Time:	2009-12-09 23:16:49	Custom Views:	Policy/Violations.Clear_Text_Application_Usage FlowShape.NearSame_Internet QoS.BestEffortObject
Source Payload 14 packets, 1006 bytes	<div> <div>UTFHexBase64</div> <div> USER PASS PWD SYST PWD PASV LIST TYPE I PASV STOR /Northwood vs. Wayne State 12-09-09.wmv </div> </div>		
Destination Payload 16 packets, 1560 bytes	<div> <div>UTFHexBase64</div> <div> 220 FTP Server ready. 331 Password required for 230 Guest access granted for 257 "/" is the current </div> </div>		

Figure 12

8. Detecting Client Side Attacks

Client side attacks are one of the top methods for a successful intrusion. Instead of an attacker targeting a server service directly, client side attacks are made possible by internal clients visiting malicious websites or content. “These are attacks that target vulnerabilities in client applications that interact with a malicious server or process malicious data. Here, the client initiates the connection that could result in an attack. If a client does not interact with a server, it is not at risk, because it doesn’t process any potentially harmful data sent from the server.” (Ridden, 2008) Successful client side attacks are usually aimed at one of two goals; either making the system part of a botnet or using the compromised system to attack other internal resources. When considering client side attacks, most people initially think of antivirus. Collecting and correlating antivirus logs is certainly a good idea. However, as previously stated, today’s attacks are regularly bypassing antivirus and this technology alone cannot be effective. Antivirus products which include heuristic or anomaly based detection may provide more valuable data as they can be correlated with other indications of compromise to isolate which system have a higher probability of being compromised. However, our focus will be on correlation of log data outside of antivirus alone.

Since the majority of operating systems continue to be Microsoft Windows based, the windows client logs are good place to start. First, the authentication and account management rules discussed in section 4 also apply to client side attacks. Creating alerts for similar activities such as locally created accounts and local group membership changes, especially local administrator’s group, are important.

Second, monitoring process information and new services can be very helpful in identifying rogue applications and malware. Windows produces an event log entry as each process starts (event ID 592/4688) and exists (event ID 593/4689). Once a client side attack has been identified, process logs can be extremely helpful in determining if any other systems inside the organization have been compromised in a similar manner. In addition, process logs may help in determining what an attacker did after the initial attack. Similarly, Windows will monitor a new service being installed. Event IDs 601 and 4697 will alert you to the installation of a new service. (Franklin-Smith)

Third, Windows scheduled tasks are another event worth monitoring. Scheduled tasks are logged in Windows using event ID 602 and 4689 “Scheduled Task Created” (Franklin-Smith). Scheduled tasks can be used by attackers to regularly schedule some kind of attack or malware update. Scheduled tasks should be monitored on both servers and client workstations. Consider developing an alert, emailed to systems administrators for server side scheduled tasks. This will allow administrators to help identify malicious actions.

Fourth, changes to the audit policy or clearing of the event log. Attackers may attempt to hide their tracks by changing the audit policy to no longer log specific event or clearing the event log after a compromise. Either of these occurrences should be considered highly suspicious and investigated. Event ID 612 and 4719 logs changes to the audit policy and event id 517 and 1102 logs the security log being cleared. (Franklin-Smith)

Beyond Windows, there are a multitude of options for logging potentially malicious activity and correlating events. Organizations just need to make sure that whatever solutions they employ or are evaluating are supported by their SIEM solution. For instance, file integrity monitoring solutions which can log appropriate changes to the file system, especially new executables and registry changes. Both are good indicators of compromise. Host based intrusion detection systems may also have some of these features. Also, consider third party systems and devices which may help in identifying compromised machines. IDS systems may have alerts detecting possible infected computers. Commercial options such as the FireEye Security Appliance or the freely available BotHunter solution are great options to integrate into your existing log activities. The more sources you can correlate with the more likely your organization can be successful in detecting these attacks.

The following example in Figure 13 ties many of these concepts together into a real world example using Qradar. The offense description shown in the first red circle is the name of the exploit detected. This exploit refers to an obfuscated PDF document. The second red circle indicates that the system is vulnerable to the exploit. While an obfuscated PDF is not by itself malicious, Qradar has gathered vulnerability assessment data on the system and knows it's running a vulnerable version of Adobe Reader. Below

the blue circle, the top ten events are listed. Based upon the information provided, it appears that the following the exploit there has been some account logon activity and group membership changes. Clicking the “Events” button (blue circle) provides a full listing of the events shown in Figure 14. This list shows various events including the original exploit, several net.exe commands being issued, a user account being created and a group membership change. Clicking the “User Account Created” event shows that an account called “haX0r” was created locally. The “Group Member Added” event shows that the recently created account has been added to the administrator group.

Offense 882

Magnitude	Relevance	Severity	Credibility
4	4	5	2

Description: HTTP:STC:ADOBE:PDF-OBfuscation

Event count: 74 events in 12 categories

Attacker/Src: [Redacted]

Start: [Redacted]

Target(s)/Dest: Local (2) Remote (14)

Duration: 4m 57s

Network(s): Multiple (3)

Assigned to: Not assigned

Notes:

Attacker Summary

Magnitude	User
[Redacted]	LOCAL SERVICE

Description: [Redacted]

Asset Name: Unknown

Vulnerabilities: [Redacted]

MAC: [Redacted]

Location: [Redacted]

Asset Weight: 0

Top 5 Local Targets

IP/DNS Name	Magnitude	Vulnerable	Chained	User	MAC	Location
[Redacted]	[Redacted]	Yes	No	LOCAL SERVICE	[Redacted]	[Redacted]
[Redacted]	[Redacted]	Unknown	No	[Redacted]	[Redacted]	[Redacted]

Top 10 Events

Event Name	Magnitude	Log Source	Category	Destination	Dst Port	Time
Misc Exploit - Event CRE	[Redacted]	Custom Rule Engine-8 : radar	Misc Exploit	[Redacted]	[Redacted]	[Redacted]
HTTP:STC:ADOBE:PDF-OBfuscation	[Redacted]	[Redacted]	Web Exploit	[Redacted]	[Redacted]	[Redacted]
Special privileges assigned to new I...	[Redacted]	[Redacted]	User Right Assigned	[Redacted]	[Redacted]	[Redacted]
Privileged Service Called Successfully	[Redacted]	[Redacted]	Privilege Escalation Succeeded	[Redacted]	[Redacted]	[Redacted]
Successful Logon	[Redacted]	[Redacted]	Host Login Succeeded	[Redacted]	[Redacted]	[Redacted]
Successful Logon	[Redacted]	[Redacted]	Host Login Succeeded	[Redacted]	[Redacted]	[Redacted]
Privileged Service Called Successfully	[Redacted]	[Redacted]	Privilege Escalation Succeeded	[Redacted]	[Redacted]	[Redacted]
Special privileges assigned to new I...	[Redacted]	[Redacted]	User Right Assigned	[Redacted]	[Redacted]	[Redacted]
Traffic Log	[Redacted]	[Redacted]	System Informational	[Redacted]	[Redacted]	[Redacted]
Global Group Member Added	[Redacted]	[Redacted]	Group Member Added	[Redacted]	[Redacted]	[Redacted]

Figure 13

Event Name	Start Time	Category	Username	Windows_ (custom)	Windows_ImageFile (custom)	Creator_Pi (custom)
HTTP-STC-ADOBE-PDF-OBfuscation	13:59	Web Exploit	N/A	N/A	N/A	N/A
Misc Exploit - Event CRE	13:59	Misc Exploit	N/A	N/A	N/A	N/A
A new process has been created	13:59	System Status	N/A	212	C:\WINDOWS\system32\wuauclt.exe	1008
Special privileges assigned to new logon successfully	13:59	User Right Assigned	NETWORK SERVICE	N/A	N/A	N/A
Privileged Service Called Successfully	13:59	Privilege Escalation Succeeded	N/A	N/A	N/A	N/A
Successful Logon	13:59	Host Login Succeeded	NETWORK SERVICE	N/A	N/A	N/A
A new process has been created	13:59	System Status	N/A	396	C:\WINDOWS\system32\wbem\wmiprvse.exe	832
A process was assigned a primary token successfully	13:59	System Status	N/A	832	C:\WINDOWS\system32\svchost.exe	N/A
Privileged Service Called Successfully	13:59	Privilege Escalation Succeeded	N/A	N/A	N/A	N/A
Successful Logon	13:59	Host Login Succeeded	NETWORK SERVICE	N/A	N/A	N/A
Special privileges assigned to new logon successfully	13:59	User Right Assigned	NETWORK SERVICE	N/A	N/A	N/A
A process has exited	14:00	System Status	NETWORK SERVICE	396	C:\WINDOWS\system32\wbem\wmiprvse.exe	N/A
A new process has been created	14:01	System Status	Administrator	1276	C:\WINDOWS\system32\cmd.exe	2016
A process was assigned a primary token successfully	14:01	System Status	Administrator	2016	C:\Program	N/A
A new process has been created	14:02	System Status	Administrator	1892	C:\WINDOWS\system32\inet.exe	1276
A new process has been created	14:02	System Status	Administrator	1792	C:\WINDOWS\system32\inet1.exe	1892
User Account Created	14:02	User Account Added	Administrator	N/A	N/A	N/A
Global Group Member Added	14:02	Group Member Added	Administrator	N/A	N/A	N/A
User Account Enabled	14:02	User Account Changed	Administrator	N/A	N/A	N/A
User Account Changed	14:02	User Account Changed	Administrator	N/A	N/A	N/A
User Account password set successfully	14:02	Password Change Succeeded	Administrator	N/A	N/A	N/A
Group member added or removed	14:02	Group Changed	Administrator	N/A	N/A	N/A
A process has exited	14:02	System Status	Administrator	1792	C:\WINDOWS\system32\inet1.exe	N/A
A process has exited	14:02	System Status	Administrator	1892	C:\WINDOWS\system32\inet.exe	N/A
A new process has been created	14:02	System Status	Administrator	1936	C:\WINDOWS\system32\inet1.exe	420
A new process has been created	14:02	System Status	Administrator	420	C:\WINDOWS\system32\inet.exe	1276
A process has exited	14:02	System Status	Administrator	1936	C:\WINDOWS\system32\inet1.exe	N/A
A process has exited	14:02	System Status	Administrator	420	C:\WINDOWS\system32\inet.exe	N/A
A new process has been created	14:02	System Status	Administrator	1944	C:\WINDOWS\system32\inet.exe	1276
A new process has been created	14:02	System Status	Administrator	2000	C:\WINDOWS\system32\inet1.exe	1944
A process has exited	14:02	System Status	Administrator	1944	C:\WINDOWS\system32\inet.exe	N/A
A process has exited	14:02	System Status	Administrator	2000	C:\WINDOWS\system32\inet1.exe	N/A
A new process has been created	14:03	System Status	Administrator	236	C:\WINDOWS\system32\inet.exe	1276
A new process has been created	14:03	System Status	Administrator	292	C:\WINDOWS\system32\inet.exe	236
A process has exited	14:03	System Status	Administrator	292	C:\WINDOWS\system32\inet1.exe	N/A

Figure 14

9. Conclusion

Attackers continue to find new methods for penetrating networks and compromising hosts. Therefore, defenders need to look for indications of compromise from as many sources as possible. Collecting and analyzing log data across the enterprise can be a challenging endeavor. However, the wealth of information for intrusion detection analysts is well worth the effort. SIEM solutions can help intrusion detection by collecting all relevant data in a central location and providing customizable alerting and reporting. In addition, SIEM solutions can provide significant value by helping to determine whether or not an incident occurred. The challenge for analysts is creating effective alerts in order to catch today's sophisticated and well funded attackers.

Those new to SIEM should start small, implementing a few of the basic methods, test them and understand their output before moving to more advanced options. Also, analysts must have the time and capability to continually review their detection mechanisms and look for new methods for detecting compromise. In the end, the goal of "SIEM Based Intrusion Detection" should be to have enough data available to the analyst to identify a potential compromise and provide as much detail as possible before beginning formal incident response processes.

10. References

- Baker, W. H., Hylender, C. D., & Valentine, J. A. (2009, December 9). *2009 data breach investigation supplemental report*. Retrieved from <http://www.verizonbusiness.com/go/09SuppDBIR>
- Beale, Jay, Baker, Andrew, Caswell, Brian, Poor, Mike, Foster, James, Beale, Jay, Baker, Andrew, Esler, Joel, Kohlenberg, Toby, Northcutt, Stephen, Rash, Michael, Orebaugh, Angela, & Turnbull, James. (2004). *Snort 2.1 intrusion detection, second edition*. Syngress Media Inc.
- Bejtlich, Richard. (2004). *The Tao of network security monitoring*. Addison-Wesley Professional.
- Bejtlich, Richard. (2006). *Extrusion detection*. Addison-Wesley Professional.
- Bejtlich, R. (2008, October 25). Security event correlation, looking back, part 3 [Web log message]. Retrieved from http://taosecurity.blogspot.com/2008/10/security-event-correlation-looking-back_4144.html
- Danford, R. (2009, January 28). *Embedded device security assessment*. Retrieved from <http://isc.sans.org/diary.html?storyid=5755>
- Franklin-Smith, R. (n.d.). *Exhaustive research on the windows security log*. Retrieved from <http://www.ultimatewindowssecurity.com/securitylog/encyclopedia/default.aspx>
- Harms, K. (2008). State of the Hack. *Proceedings of the NEbraskaCERT Conference 2008*, <http://www.certconf.org/presentations/2008/files/C4.pdf>
- Kamluk, V. (2008, May 13). *The Botnet business*. Retrieved from <http://www.viruslist.com/en/analysis?pubid=204792003>
- Kotadia, M. (2006, July 19). *Eighty percent of new malware defeats antivirus*. Retrieved from <http://www.zdnet.com.au/news/security/soa/Eighty-percent-of-new-malware-defeats-antivirus/0,130061744,139263949,00.htm?omnRef=1337>
- Mookhey, K. (2004, March 17). *Detection of sql injection and cross-site scripting attacks*. Retrieved from <http://www.securityfocus.com/infocus/1768>
- Netwitness total network knowledge siemlink. (2009). Retrieved from <http://www.netwitness.com/products/SIEMLink.aspx>

- Nicolett, M., Heiser, J., Colville, J., et al. (2009). *Hype cycle for governance, risk and compliance technologies, 2009*. Retrieved November 22, 2009, from Gartner database
- Ridden, J. (2008, August 16). *Client side attacks*. Retrieved from <http://www.honeynet.org/node/157>
- Sans: top twenty security problems, threats and risks*. (2007, November 28). Retrieved from <http://www.sans.org/top20/>
- Shadowserver foundation information botnet detection*. (2009, July 25). Retrieved from <http://www.shadowserver.org/wiki/pmwiki.php/Information/BotnetDetection>
- Source security event 528*. (n.d.). Retrieved from <http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows%20Operating%20System&ProdVer=5.0&EvtID=528&EvtSrc=Security&LCID=1033>
- The Darknet project - team cymru*. (n.d.). Retrieved from <http://www.team-cymru.org/Services/darknets.html>
- The Truth about false positives*. (2001). Retrieved from <http://documents.iss.net/whitepapers/TheTruthAboutFalsePositives.pdf>
- Ullrich, Johanness. (2009, June 4). My Top 6 honeytokens [Web log message]. Retrieved from <http://blogs.sans.org/appsecstreetfighter/?s=web+honey+token>