Technical Paper

The Nexpose Expert System

Using an Expert System for Deeper Vulnerability Scanning

Executive Summary

This paper explains how Rapid7 Nexpose uses an expert system to achieve better results in vulnerability scanning compared to traditional procedural methods. After a brief discussion of the product objectives and implementation, we will take an in-depth view of the use of expert systems to achieve accurate and detailed vulnerability results.

The detection of vulnerabilities in a computing system relies on the simultaneous processing of multiple events. Asset discovery detects assets while fingerprinting attempts to identify systems, applications, services and databases. Vulnerabilities may be exploited to expand the range of systems that may be detected, and vulnerabilities may be exploited on one system to penetrate other systems.

Nexpose is able to handle all these cases and many more. Nexpose operates an intelligence gathering expert system designed to penetrate systems in both known and unpredictable ways.

Significant and unique to Nexpose is an expert system called *Jess* that can use exploits to provide feedback to access more rich information about assets and vulnerabilities. This is similar to the mechanisms used by real hackers because it builds on prior knowledge. Examples of this layered architecture are provided in this paper.

This paper is relevant to security professionals interested in better techniques of finding vulnerabilities who have a solid understanding of networking principles and familiarity with the concepts related to hacking, vulnerabilities, and exploits. Some programming knowledge and/or experience with Nexpose is useful but not necessary.

RAPID7

Objectives of Vulnerability Management

Vulnerability management is the process of cataloging vulnerabilities in a network environment and then determining which of those vulnerabilities present unacceptable risks that require remediation. Comprehensive vulnerability management solutions, such as Nexpose, also include advanced trend reporting, management of asset data, ticketing and configuration compliance scanning in order to get a holistic picture of risk related to vulnerabilities and compliance issues in the environment.

One of the biggest challenges for any vulnerability management program is the analysis of scan results, which must be verifiable and actionable in order to effectively remediate them. With some solutions, customers can be overwhelmed with false positives which can affect the credibility of the security team. The high incidence of false positives forces organizations to spend a considerable amount of time verifying scan results prior to fix coordination. Over time, false positives can lead to "warning fatigue," undermining the credibility of the security program. This is the primary roadblock for organizations hoping to automate the link between scanning and remediation. Also problematic are false negatives, which can give an organization a false sense of security.

Some of the complexities a vulnerability management solution must overcome include:

- Many exploits are not safe and can disrupt production systems if not used correctly.
- Authenticated access may not be available or permitted which can lead to less accurate information on patch levels for applications.
- Hardened systems may have missing or inaccurate banners that cannot be relied upon for fingerprinting.

Nexpose was designed to solve many of these problems by incorporating an expert system and abstracting the scanning complexities. This approach allows the building of a simpler vulnerability check model with a higher degree of accuracy. Good knowledge provides real risk analysis, credible remediation plans and easy to use data management functions.

Nexpose Architecture

The objective of vulnerability scanning is to provide accurate and detailed vulnerability information related to the set of assets scanned. Nexpose does this by leveraging known vulnerabilities and default configurations to access data and demonstrate how control of the system can be achieved. Nexpose also offers the ability to dial up or down the level of assessment and penetration to provide the most accuracy with no effect the reliability of the scanned systems.

Nexpose is able to perform policy and vulnerability checks against network systems, operating systems, web applications and databases from a single scan and correlates the results using the built-in expert system to provide proof of attack vectors.

Nexpose operates in a similar manner to a cyber-criminal, using information discovered in one domain to reveal vulnerabilities in others. An example is the discovery of a database password that allows access to the registry of the database host server. The registry identifies the software levels of all installed software, and Nexpose uses that information to attack the vulnerabilities exposed. Knowledge of these vulnerabilities could provide access to the root of the server thereby allowing complete control of the server and thereby access to the entire database. The section on *Jess* highlights how these feedback mechanisms are used to propagate attacks.

Nexpose Components

Nexpose has two components, a central server, and one or more scanning engines. The central server is called the NSC (Nexpose Security Console) and the scan engine is called NSE (Nexpose Scan Engine).

The central server runs a Web server process to interact with its users, a backend database for information storage and a scan engine to scan assets. Additional scan engines can be placed strategically in the network to originate scanning under the control of the NSC.

The architecture is distributed, with engines and servers communicating over SSL connections. The sensitive nature of the data requires authentication performed by a hierarchical Public Key Infrastructure. This implementation secures local communications and also allows the use of common engines located at Rapid7's data center to perform external scanning on behalf of many clients while maintaining security and data integrity.

Users of Nexpose log on using a web browser and an administrator-provided username and password through the NSC home page. Depending upon the level of access granted, users may be restricted in their permissions to view or scan assets. Administrative activities include establishing scanning schedules, running scans, creating reports and reviewing remediation activities.

Scanning

The scanning of systems is typically run through three phases. In the first phase, asset discovery is performed. This identifies the assets that are responding to requests, and starts service fingerprinting.

The service fingerprinting phase attempts to identify what services are being provided by the computing system so that the subsequent vulnerability detection phase can use those fingerprints to launch attacks using the appropriate vectors. For example; SSH vulnerability detection should only be performed against the appropriate ports and Windows vulnerability checks are limited to Windows systems only.

Vulnerability and policy compliance detection occurs throughout the scanning phases, and as the scan progresses higher level vulnerabilities are detected and reported in web applications, databases and custom applications as well.

Reporting and Notification

While Nexpose has extensive reporting and notification capabilities, these are not discussed extensively in this paper, except where it relates to the *Jess* engine. Contact Rapid7 for any information about Nexpose not covered in this document, or visit the Rapid7 website at www.rapid7.com.

RAPID7

Jess in Nexpose

The central component of the Nexpose scanning engine is the expert system, *Jess*. *Jess* is the Java Expert System Shell and has been used by Rapid7 since 2000. Most of the scanning infrastructure code is written in the *Jess* language, which is similar to LISP and Scheme. Online *Jess* documentation is available at http://www.Jessrules.com/Jess/docs/for those interested in programming concepts.

The two primary components in an expert system are rules and facts. Facts constitute the data that is managed by the expert system. In Nexpose, facts represent information discovered during the scan, such as asset status, open ports, and fingerprints. Rules match against one or more facts and execute code when all required conditions are met. For example, to start a service discovery with port scan the asset discovery engine sets the alive fact. A rule triggers on assets with status set to alive. When an asset is encountered with its status set to alive, the rule will be activated and the port scan will be run against the asset starting service discovery. In *Jess* parlance, the trigger conditions are called the left hand side (LHS) and the actions to execute are the right hand side (RHS).

Information Gathering Techniques

Nexpose defines an abstract service model allowing *Jess* rules to leverage information gathered during a scan. The existence of a service capable of feeding information into the knowledge base is asserted as a *Jess* fact such as "RegistryServices" indicating that the registry can be read on an asset through some discovered vulnerability. Rules can then make use of these various services to perform their checks, and potentially advertise additional services based upon their results.

"Jess is 'the real programmer's rule engine,' designed with folks like Rapid7 in mind. Their innovative use of rule technology lets Nexpose handle an enormous number of complex scenarios within a clean and elegant structure. The modular nature of rules means that new scenarios can be added easily without compromising the integrity of the database while Jess's high performance execution engine helps keeps everything running quickly."

- Ernest Friedman-Hill, Advanced Software Research, Sandia National Labs - Jessrules.com

By way of example consider file access. There are many vulnerability and policy checks that require the ability to read files. The file size, version and header information helps validate software versions. Configuration information is stored in files, and in some cases passwords can be retrieved from files. Certainly an unauthorized cyber-criminal will attempt to remove files for inspection to launch further attacks where they can be found.

Nexpose vulnerabilities often want to read files and so if any of these vulnerabilities are found to exist they will be internally advertised to the Nexpose vulnerability scanning engine as generic File Reading services. Any Nexpose vulnerability check that needs file access will be passed the file. The file components required will be retrieved and made locally available on the scan engine for the duration of the scan to provide information to additional checks requesting file information. An example of this would be a database configuration file that can provide both the default administrative credentials as well as server configuration information for policy reporting.

Emergent Behavior

Nexpose, through the use of *Jess*, can display intelligent behavior and chain complex mechanisms together to drive towards a defined result, known as goal driven processing. In an expert system, backward chaining allows the effect to drive the cause. In other words, the need to get file access drives vulnerabilities to create file access. This occurs in Nexpose in different and unanticipated ways and leads to emergent behavior that finds more than one would expect.

When Nexpose detects a restricted service that fact triggers escalation rules that attempt to upgrade the service to an unrestricted service running with administrative privileges. As soon as administrator privileges are established, Nexpose starts running local system checks and detailing missing system patches.

It is not always possible to predict the paths that Nexpose will take to scan a network environment and services can build upon other services to demonstrate behavior not seen before. This emergent behavior is how Nexpose can find things that other scanners are not able to find.

An Example

The real power of Nexpose's generic service capability model is that vulnerability checks can be written in a way that is agnostic as to the means by which they are performed. This allows vulnerability checks to be performed using any combination (or "vector") of services obtained during a scan. The following real-world example is taken from a customer environment.

Assume the following scenario. A Nexpose administrator is responsible for performing authenticated vulnerability scanning of a Windows network, however, some of these machines are rogue and not integrated with Active Directory. Therefore, the scan credentials provided by the Nexpose administrator will not work. The rogue machines could be misconfigured and could be missing critical security updates, which presents a real risk to the environment.

Here is how the expert system in Nexpose handles this situation:

1. The assets are discovered to be alive during asset discovery.

2. TCP ports 135, 139, and 445 are discovered open during service discovery.

3. The ports from step #2 are fingerprinted and determined to be for WMI and CIFS which are two administrative services present on most Windows assets.

4. By inspecting the fingerprinting results from #3 as well as IP stack fingerprinting, Nexpose is able to determine that these are Windows assets, however, the exact version may not be known yet.

5. Because the administrative credentials do not work on these rogue machines, no software, users, etc. are enumerated.

6. Vulnerability checking commences, however, the only types of checks that run and return anything are those that don't require administrative credentials. Among the checks that run is a check to ensure that the password for the Administrator account is not a default password. The host is found to be vulnerable to one of the default passwords checked by Nexpose.

7. Because step #6 discovered valid credentials, these credentials are then made available to anything which can use them. This means that the expert system can now run step #5 to enumerate software and users on the rogue asset.

8. The checks run and find several high-profile, high-risk Internet Explorer vulnerabilities which presents a serious risk to the organization.

Vulnerability Exploits

Nexpose can also exploit vulnerabilities to provide services ranging from file access to remote execution. The results of the exploits are either confirmed or unconfirmed. In the event that the vulnerability is confirmed, then it has been proven and the exploit information is provided as proof.

Nexpose and Metasploit, Rapid7's penetration testing solution, are integrated closely together to verify vulnerabilities are exploitable which helps remediation teams determine which vulnerabilities to remediate first.

RAPID7

Asset and Service Discovery

The global IP address space covers billions of assets broken into practical blocks of hundreds or thousands of individual addresses over the IPv4 and IPv6 address spaces. Discovering the assets within the corporate address space requires a high speed scan of a large address space to identify all responding assets.

When scanning assets, identification of responding assets requires high speed, short packets and efficient multitasking to keep the network responsive. Nexpose allows tailoring of the scan in several ways. Traditionally a ping was used to identify a responding host computer. This may work under some circumstances, but most likely either a firewall or intrusion prevention system will ignore the packet to discourage scanning. If the owner of the system has taken action to mask the existence of the system then the task becomes more time consuming and complex.

Firewalls often have rules allowing HTTP protocols over port 80 to pass unimpeded which can allow a TCP ACK to be sent to a web server and for a response to be received. This is the most common technique for probing through firewalls.

In more complex cases, only a specifically crafted request to a non-standard port might work, or the owner of the system may have configured the system to respond to only certain IP addresses, in which case it is unlikely that the asset can be discovered at all.

The asset discovery phase provides *Jess* with facts that drive the service identification rules. As facts about ports are set, the *Jess* rules start trying to authenticate with the services to attempt authenticated logons.

Fingerprinting

Fingerprinting is the accurate detection of operating systems, applications, and services in order to get a complete picture of the state of the asset and vulnerabilities that are present. Fingerprinting occurs throughout a scan and constantly seeks to use information discovered to upgrade the current fingerprint. Nmap ("Network Mapper") is a free open source utility for network exploration or security auditing. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. Nmap is free and open source and is found at http://insecure.org/nmap/.

Nexpose uses Nmap for Asset and Port Discovery. Nexpose provides many facilities above and beyond Nmap such as service fingerprinting, connections to database servers, RPCs, SSH, web servers, SNMP fingerprinting and many more. The fingerprint model within Nexpose is designed to make use of the best fingerprint for a service, especially when many conflicting fingerprints may exist. This is done by taking into account the reliability of the source (e.g. Windows registry is far more reliable than a service banner, and weighted accordingly) and the specificity of the fingerprint (e.g. Windows 2000 Advanced Server SP1 is more specific than Windows 2000, and will be weighed higher for equivalent reliabilities). A certainty value (0.0 - 1.0) is then assigned to the fingerprint, which can also be used within vulnerability checks (minCertainty attribute) to help prevent the usage of low certainty fingerprints. As the knowledge base grows for an asset, the accuracy of information is refined.

Service Identification

The service identification phase is used to map out the network services running on the active assets, to continue asset identification and to attempt intrusion using network and OS vulnerabilities.

Service Discovery can be tuned to enable or disable TCP and UDP port scans. Specific ports can be specified for scanning, as well as default port lists or all possible ports (1 - 65,535). Additionally, the method of TCP port scanning can be changed to use full connections, half-open (SYN) scans as well as other variations.

Once a port is deemed to be open, Nexpose performs a protocol handshake on that port to verify the type of service running on the port. This allows Nexpose to determine if a service is running, even if it is not the expected port. For example, an HTTP server may be running on port 1234, which is not the standard HTTP port 80.

Network vulnerabilities have the capability to expand the range of discovered assets. For example due to ambiguities in TCP/IP implementations, it is sometimes possible to bypass firewall rules intended to keep state on outbound connections. If multiple conflicting flags are set on the initial SYN packet sent in the TCP/IP handshake, some operating systems allow the SYN packet to pass even if they are configured to block incoming connections. This allows Nexpose to conduct asset discovery through a vulnerable firewall to find new assets and services.

Vulnerability and Compliance Detection

Vulnerability Detection is based on proactive scanning of systems and services. Nexpose uses the network, OS and services layers identified during discovery and fingerprinting phases to its full advantage for subsequent vulnerability detection. During the vulnerability detection phase, Nexpose further explores the target environment by scanning system, web and database internals.

Using the expert system, Nexpose is able to identify known vulnerabilities and configuration compliance issues across web sites/services, databases, network equipment, operating systems, and applications. All of this detection can occur during the same scan window from the same scan engine which makes it simple to configure and simple to get all the information you need at one time to assess risk in the environment.

The rules engine continues to execute rules based on the changing facts in working memory until there are no rules available to execute, signifying the end of the scan. Nexpose notifies users of progress in the scan and users can see results as they are happening during the scan.

Remediation Reporting

Once the vulnerability and compliance scanning has completed, users can now assess the risk and determine what is most important for their environment. Nexpose includes several reports which help with this including the Prioritized Remediation Report, Top 10 Vulnerability Report, and Audit Report. The Prioritized Remediation report takes into account all available patches and all known vulnerabilities in the environment and provides a prioritized list of which remediations will have the most impact on risk in the environment. This saves time as the report is actionable with clear remediation steps and the detailed asset information required to remediate the risk.

Once you have established a baseline for the environment, the Vulnerability Trend reporting can then provide information on how the environment is changing over time. For example, it is important to know when new assets are being added to the network, when new vulnerabilities are being found, and when older vulnerabilities are not being remediated.

Nexpose also offers the ability to report on the assets and vulnerabilities that are important to the environment using rich asset and vulnerability filtering. All of these reports can be automated from the UI or API so that as soon as a scan completes, remediation owners get the accurate and detailed information they need to do their jobs and stakeholders can get accurate information on how risk is changing over time.

About Rapid7

Rapid7's security solutions deliver visibility and insight that help you make informed decisions, create credible action plans, and monitor progress. They simplify risk management by uniquely combining contextual threat analysis with fast, comprehensive data collection across your users, assets, services and networks, whether on premise, mobile or cloud-based. Rapid7's simple and innovative solutions are used by more than 2,500 enterprises and government agencies in more than 65 countries, while the Company's free products are downloaded more than one million times per year and enhanced by more than 200,000 members of its open source security community. Rapid7 has been recognized as one of the fastest growing security companies by *Inc. Magazine* and as a "Top Place to Work" by the Boston Globe. Its products are top rated by Gartner® and *SC Magazine*.

For more information about Rapid7, please visit http://www.rapid7.com.