

Summary

The recent introduction of the TIssue baseline detectors has brought a request for a new type of exemption – a TIssue issue exemption (also called a baseline variance). This new exemption request has caused a review of the current “scanning exemption” architecture. The current exemption architecture cannot accommodate the requested new exemption type. In order to implement a new exemption type, the best approach is to re-architect the existing “scanning exemption” architecture.

Introduction

Since periodic network scanning started in late 2001, exemptions from enforcing security policies have been needed to accommodate various network devices. Initially, exemptions were implemented ‘ad hoc’, for example as a white list in a particular scanner. As the number and importance of exemptions increased, the CST-RA (Computer Security Team – Risk Analysis) application was developed in part to organize and maintain exemptions.

The CST-RA has not been deployed in production, so the management of exemptions still lacks a user interface and work flow (e.g. exemption notifications). With the addition of new TIssue issues that can create network blocking events, the management of exemptions is even more important. Users need a way to view existing exemptions, to be notified of existing exemption expiration, and to request/renew exemptions. Since TIssue builds events for detectors that find issues, this proposal is to use TIssue to manage all exemptions. The TIssue Exemption Manager is the proposed solution for handling exemptions for all scanners and TIssue detectors.

Overall Description

The following applications have been developed as an enforcement aid for the Computer Security policy. These applications are collectively referred to as NIMI – Network Information and Management Interface.

- **Inventory DB** – This is a near-real-time inventory of all active network devices and their associated services. A network device is considered to be any device that is connected to the Fermilab network. Network devices offer services on various ports.
- **NIMI DB** – This is a near-real-time collection of network data organized in tables such as ARP/Switch, DHCP lease, temporary network registration, etc. The NIMI DB is useful to resolve network devices to particular network hardware and configurations (e.g. virtual LANs) and to timely notify scripts (via the NIMI Event Dispatcher) when a network device changes state (e.g. goes on-line).

- **Tissue** – This is an issue tracking application. Issues are defined in Tissue and when an issue detector finds a system with a defined issue, a Tissue event is created for tracking. A workflow is available to notify registered users of event creation and for event reminders. A user GUI is available for users to interact with events by remediating the issues found.

Definitions

- **network device** – any device connected to the Fermilab network. These devices have been called systems, nodes and machines. This document uses the term ‘device’ to refer to all of systems, nodes and machines connected to the Fermilab network.
- **port scanning** – activity used to populate/update the NIMI Inventory DB. Port scanning uses the nmap tool to find devices and to determine the services offered by a device. The parameters applied to the nmap tool control the scope (i.e. number of services searched) and intensity (i.e. service detection) of the port scan. For devices sensitive to port scanning, exemptions are needed to control the nmap parameters.
- **SA scanning** – Strong Rule Authentication scans used to detect non-Kerberized network services. These scanners are homebrew.
- **VS scanning** – Network vulnerability scans are used to detect system software vulnerabilities in services that can be exploited via a network connection. These scanners mostly use the nessus scanning tool.
- **RA scanning** – Remote access tests used to enforce policy for accepting/rejecting remote connections. These scanners are homebrew.
- **Issue detectors** – Issues are defined in the Tissue DB. Detector scripts are developed mostly by the Security team to look for policy violations or suspicious activities (either network or user).
- **Issue tracking** - When an issue is found by a detector, the Tissue application is used to track the issue. Detectors report results to Tissue using the detector’s unique “source class”. The detector source class ensures issues defined for one detector do not collide with issues defined for other detectors (i.e. each detector has it’s own namespace).
- **Services** – A network service is a piece of software running on a device that listens on a network port for connections with the purpose of providing resources to a client . In this document, services refer to either the network port number used or the name of the service. For example, the secure shell service usually runs on network port 22 and is named ‘SSHd’. Issues are typically defined with relation to the service name rather than the network port. The particular detector translates the network port into a service name.
- **Issue** – One of a list of computer security policy violations. Each issue defined in Tissue has a unique “issue_code” within a detector source class. A device with an identified issue is tracked by creating a Tissue “event”. The Tissue event combines the device, issue and time stamp. The event is tracked using the Tissue work flow

NIMI Overview

The operation of NIMI is as follows:

1. Software scripts known as “collectors” access routers and switches (specialized network devices) to collect the identifying data for on-line devices. These scripts analyze the data and populate tables in the NIMI DB.
2. Software scripts using recent data from the NIMI database and from “ping sweeps” of the network identify devices newly connected to the network. Other scripts access the Inventory DB to identify on-line aged devices that need a current port/services scan. Jobs are created for the Fermi Scanner Farm (FSF) to execute port/service scans against the identified list of devices. The results of the port/service scans are used to update observations and services in the Inventory DB.
3. Detectors are used to test policy compliance. For example, scripts look into the Inventory DB for new or recently refreshed devices that have “interesting” services open. These devices are selected to test for compliance using other software scripts commonly referred to as “scanners”. Jobs are created for the FSF to execute the “scanner” scripts. Other data sources (e.g. syslog logs) are searched for devices found to be out of compliance for various reasons (e.g. failed virus scan or old kernel software). Detectors scrape these sources for known issues.
4. When a detector finds a device that is not in compliance with one or more security policies, the device is assigned an issue by the detector. The detector then communicates with TIssue to create a TIssue event. The combination of network device identification, detector issue and time stamp creates a TIssue event that is tracked.

Exemption Overview

For systems with known and approved security policy variations, an exemption process is needed to modulate the port scanning activity and the creation of events in TIssue. Here are the classes of exemptions that are required:

1. **Port Scan Exemptions** – Step 2 in the “NIMI Overview” above describes how port scans are used to maintain the Inventory DB. For network devices that are sensitive to a port scan, the nmap parameters need to be adjusted. The existing port scan exemptions are called “system” exemptions. Several have been defined:
 - NO_S** Prevent all network scans. This is used for very sensitive network devices.
 - NO_64** Prevent complete (64K) port scans. This is used for network devices that have some sensitive network services.
 - NO_A** Prevent service detection scans. This is used for network devices where all network services are sensitive to detection probes.

SYS Prevent vulnerability detection scans.

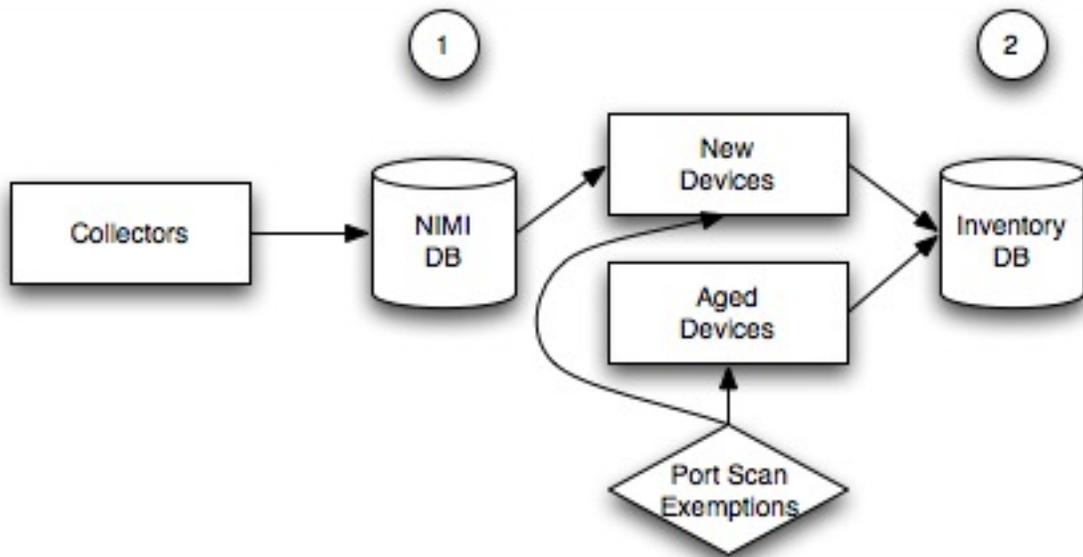
DY64 Prevent complete (64K) port scans. This is a temporary exemption for devices that exhibit a scanning timeout. This exemption will not use the “exemption work flow”.

2. **Service Exemptions** – Step 3 in the “NIMI Overview” above describes how specialized scanners are used to detect issues. For devices that are sensitive to these specialized scripts, device service exemptions may be needed. Note, these exemptions to the detection (scanning) script do not currently exist.
3. **Issue Exemptions** – Step 4 in the “NIMI Overview” above describes how a detector creates TIssue events. For network devices that have known variances from security policy, the creation of a TIssue event needs to be suppressed. Note, issue exemptions do not currently exist.

Referencing the steps above in the NIMI Overview, here is where the Port Scan Exemptions are used.

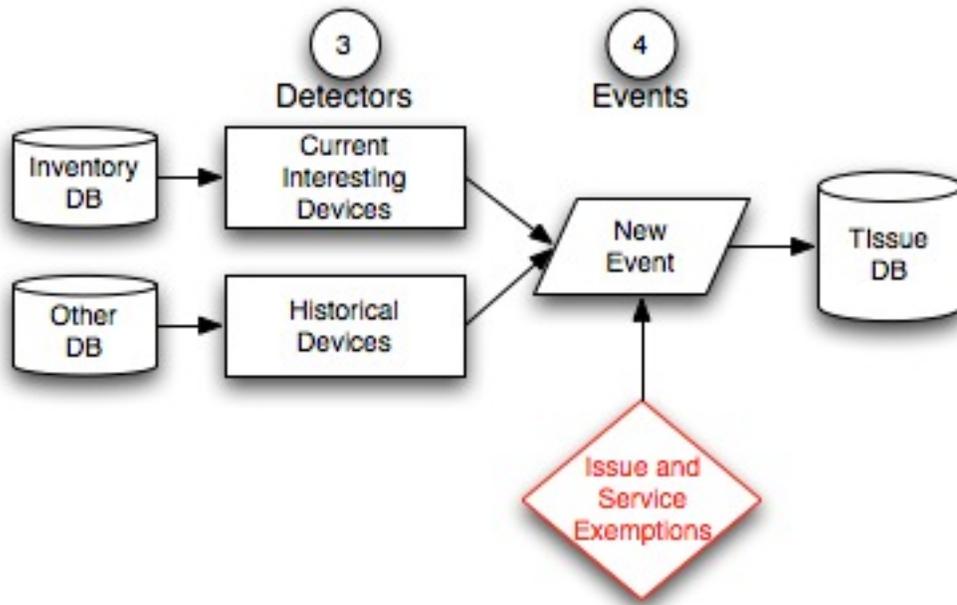
Exemption Overview – Port Scan Exemptions

Port scanning affects the entry of new (i.e. recently connected to the network) and aged devices into the Inventory DB.



Exemption Overview – Service and Issue Exemptions

Service and issue exemptions affect the TIssue event creation process.



System Features

The TIssue Exemption Manager software should have the following features:

1. Users and approvers use a web interface to maintain exemptions.
2. The TIssue Exemption Manager will support a command line interface. The command line interface will likely be useful for bulk changes such as extending exemption expiration dates.
3. Access to the Exemption Manager is authorized via a KCA certificate.
4. Authorization for the Exemption Manager will use defined roles.
5. Exemption workflows provide for:
 - a. email notifications,
 - b. email approval notifications,
 - c. discussions related to exemption requests,
 - d. email exemption expiration notification.

There may be more than one workflow defined.

6. The exemption information is available to software scripts over the network (e.g. via a web service).
7. An exempt network device must be registered in MISCOMP.
8. Each device has a unique identifier – (e.g. MISCOMP sysID or MAC address).
9. A particular device can have one or more exemptions.
10. Exemptions remember requestor and approvers (likely an approver chain?).
11. Exemption have information about exemption type, exemption code, request date, expiration date, approved, approval date.

12. The TIssue Exemption Manager knows about clusters of devices (e.g. a farm or grid). Clusters can be defined in the manager or can be imported from MISCOMP.

System Architecture

The TIssue Exemption Manager should have:

1. KCA for authentication
2. Roles (based on KCA authentication) for authorization
3. A command line interface
4. An exemption database
5. Forms
 - a. requests and renewals
 - b. reviews and discussion
6. Activities
 - a. approval, requests, rejection, renewal
7. Reports
 - a. List of all exemptions for a given requestor, approver, device.
 - b. List of expiring exemptions for a given requestor.
 - c. List of expiring exemptions in next N days.

External Interfaces

User Interfaces

Device Identification

A user needs to identify a particular network device or group of devices. The methods used to identify a device/group can be (one or more of):

- Name (aka node or system name),
- IP address (if device has a IP addresses assigned),
- MAC address,
- Cluster definition (e.g. locally defined or imported).

It may be easiest for users to browse network devices “assigned” to them using:

- User Name,
- Fermi ID,
- Cluster name

Exemption Selection

Once one or more network devices are identified, a list of available exemptions can be browsed. The user can select one or more devices and one or more exemptions and then submit a request for approval. A default expiration of one year will be suggested. A default reason will be highlighted (for example in a radio box).

Reasons are “new”, “delete”, “renew”. A free-form comment is required for the user to explain any additional information for the request.

Workflows

The Exemption Manager will use the following workflows.

1. Request Workflow
Once an exemption request is submitted, a work flow is invoked that will:
 - a. Summarize the request in an email to the requester,
 - b. Notify the approver of the request (i.e. invoke the approval workflow),
2. Approval Workflow
Once an request/renewal is submitted, a work flow is invoked that will:
 - a. Track if the request is approved or rejected (i.e. invoke the discussion workflow),
 - b. Notify the user of the request status (approved, rejected, expired)
3. Discussion Workflow
4. Expire/Renewal Workflow

APIs

Various scripts in the NIMI application will need to query for existing exemptions. The NIMI scripts, especially the scanning scripts, operate at network layer 3 aka the IP layer. Hence, the network devices being scanner or being checked for issues will be identified by an IP address.

Possible external calls/methods:

```
request_exemption(...)
```

Start the request workflow.

```
create_exemption(...):
```

Create the requested exemption in the “pending approval” state.

```
revoke_exemption(exemption_type, deviceID):
```

Delete the exemption.

```
get_exemptions(exemption_type, list_of_ips):
```

This API will return a list of IPs that have an approved, unexpired exemption of the requested type.

```
lookup_exemptions(IP_address):
```

This API will return a list of all approved, unexpired exemptions for the given IP.