CVE-2011-3872 Remediation Toolkit -- Detailed Walkthrough

This module will help you permanently remediate the CVE-2011-3872 AltNames vulnerability.

How the Vulnerability Works

Puppet agent identifies the puppet master by comparing the puppet master DNS name it knows with the names in the master's certificate. The master's cert can include both a primary name (Subject CN) and a set of alternate DNS names (Subject Alternative Name).

Alternate DNS names are optional for master certs, and have to be specifically enabled with the `certdnsnames` option. **In versions prior to 2.6.12 and 2.7.6, the Puppet CA will improperly insert any `certdnsnames` values into *agent* certificates as well as master certificates.** This bug was introduced in Puppet 0.24.0.

This means that if the following two conditions are both met:

- Your puppet master has ever had its `certdnsnames` setting turned on during the current CA's lifetime
- Any of your agent nodes are configured to contact the master at a DNS alternative name that has ever been included in the `certdnsnames` setting

...then your site probably contains agent certificates that can impersonate the puppet master in a man-in-the-middle attack. You can quickly check for such certificates with the included `bin/scan_certs` and `bin/webrick/scan certs` scripts, or you can examine any individual certificate by running `openssl x509 -text -noout -in <certificate pem file>` and looking for the X509v3 Subject Alternative Name field.

The threat posed by these certificates will persist even after upgrading to an unaffected version of Puppet. This module uses Puppet to help you to quickly neutralize these certificates.

How the Fix Works

Since two conditions must be true for your site to be vulnerable, you can protect yourself by breaking either condition:

- If you configure all agent nodes to reach the master at a new, "clean" DNS name, they cannot be spoofed by rogue agent certs with the old DNS names.
- If you migrate all machines to a new CA which has never had `certdnsnames` turned on, every dangerous certificate will be permanently invalidated.

This module can break both conditions of the vulnerability.

Steps 1 and 2 will secure your site immediately by configuring all agents to use a new DNS name for the puppet master; they will also turn off the `certdnsnames` setting if it isn't already deactivated.

Steps 3 through 5 will provide long-term protection by migrating all machines to a new CA. This will "clean" and restore your puppet master's previous DNS name.

Repair Options

You have three main options for remediating the AltNames vulnerability.

1. If you have a **small-to-moderate number of nodes and can trivially SSH to all of them,** you can use a simpler remediation workflow. Disregard the rest of this guide and consult README-ssh-only.markdown.
2. If mass SSH is impractical and you **don't mind permanently changing the puppet master's DNS name,** you can protect yourself by running only the first two steps of this module. Continue reading for instructions, and stop after step 2. You may need to modify your new node bootstrapping process to use the new DNS name.
3. If mass SSH is impractical and you **wish to continue using the current DNS name(s),** (or if you just want long-term protection against accidental reuse of the old names) you should run steps 1 through 5 of the

remediation module. Continue reading for instructions.

Walkthrough

To remediate your site with this module, you must:

- Stop adding new nodes
- Create a new temporary DNS entry for the puppet master
- Install the module
- Ensure that your own modules will not interfere with the remediation
- Run steps 1 and 2 to secure your site
- Optionally, run steps 3 through 5 to "clean" the puppet master's previous DNS names

Stop Adding New Nodes

You should not add nodes to your Puppet infrastructure during the remediation process, as it may strand new nodes in an "orphaned" state that requires manual repair.

Create a New DNS Entry for the Puppet Master

This module secures your site by configuring agents to contact the master at a temporary (or permanent) new DNS name. Before it can do this, you must choose a name yourself and edit your site's DNS configuration to point it at the puppet master. Configuring DNS is beyond the scope of this document.

If your site's change-management policies do not allow timely modification of DNS records, you can use Puppet itself to add a host entry on every agent node. Simply add a host resource like the one below to your site.pp manifest (outside of any node statements), and allow every agent to run once.

```
host {'puppetmaster.new.domain.com':
  ensure  => present,
  ip      => '172.16.158.132',
  comment => 'Temporary puppet master hostname for remediating CVE-2011-3872'
}
```

You may wish to log in to a few agent nodes and test that the new name resolves correctly.

Install the Module

This module must be installed in Puppet's `modulepath` before you can use it. If you have the puppet-module tool installed, you can run the following:

```
cd /tmp
wget http://links.puppetlabs.com/puppetlabs-cve20113872-0.0.1.tar.gz
cd $(puppet master --configprint confdir)/modules
puppet-module install /tmp/puppetlabs-cve20113872-0.0.1.tar.gz
```

If you're running an older version of the puppet-module tool, you may need to:

```
mv puppetlabs-cve20113872 cve20113872
```

If you are not running the module-tool, you can simply unarchive the tarball, rename the directory to `cve21003872`, and move it to your modules directory.

Avoid Interference

This module makes changes to the following files on every puppet agent node:

- The main puppet.conf configuration file
- The contents of Puppet's `ssldir` (run `puppet agent --configprint ssldir` to locate this directory)

If you are using Puppet to manage any of these files -- that is, managing Puppet *with* Puppet -- you **must** add a `noop => true` metaparameter to all such resources until the remediation is complete. After the remediation, you must ensure that your resources won't undo any permanent changes made by this module before turning them back on.

Step 1

This step:

- Turns off the master's `certdnsnames` setting, if it hasn't already been turned off.
- Issues a new certificate for the puppet master. This certificate will contain all of the previous DNS names for the puppet master, with the addition of a new DNS name of your choice.

This step modifies only the puppet master. You can perform the next step immediately.

PE Users

From the top directory of this module, run the following:

```
bin/pe_step1_enable_intermediate_dns_name <new DNS name>
```

Other users

Important note: Our initial set of non-PE scripts require that you run puppet master under WEBrick for the duration of the remediation process. This is because most production-scale master configurations terminate SSL at the load balancer or web server, and we were unable to automatically adjust these certificate settings in a one-size-fits-all manner. If your site operates at an extreme scale, you may need to manually tailor the tasks in each step to your master configuration, but we are hoping that most users can absorb the temporary performance hit. If you are able to automate remediation under other puppet master configurations, please consider making a public fork of this module or submitting a pull request.

- If you can maintain a secondary shell session to the puppet master server, you can start a WEBrick master with `puppet master --no-daemonize --verbose` and stop it with ctrl-C.
- If you prefer to only maintain one shell session, you can start a WEBrick master with `puppet master` and stop it with `kill $(cat $(puppet master --configprint pidfile))`.

Stop your normal puppet master. Be sure to both stop the puppet master processes and disable any supervisor process that manages the master. (For example, if you are running puppet master under Passenger, you should either stop Apache/Nginx, or restart the web server after moving the master's vhost config file to a disabled location.)

From the top directory of this module, run the following:

```
bin/webrick/webrick_step1_enable_intermediate_dns_name
```

Do not re-start the puppet master; proceed directly to step 2.

Site status after running step 1:

- CA will create dangerous certs? **NO.** (fixed!)
- Agents can be spoofed by agent certs? **YES.** (not fixed)
- Potentially dangerous certs are still valid? **YES.** (not fixed)
- Agents can operate normally and receive catalogs from master? **YES.** (business as usual)

Step 2

This step:

- Adds the `cve20113972::step2` class to all agent catalogs. This class:
  - Configures each agent node to contact the puppet master at the new DNS name.

This step modifies agent nodes. **All agent nodes must run once before performing the next step.**

PE Users

From the top directory of this module, run the following:

```
bin/pe_step2_configure_agents_for_intermediate_dns_name
```

You must now wait for every agent node to check in.

Other Users

From the top directory of this module, run the following:

```
bin/webrick/webrick_step2_configure_agents_for_intermediate_dns_name
```

Start a WEBrick puppet master server as described in step 1. If you do NOT intend to run steps 3 through 5, re-start your original production-scale puppet master.

Checking agent status

This step is not complete until every agent node has run once. You can use the included `check_progress` script to view a summary of your nodes' progress through this step; simply run:

```
bin/check_progress
```

or:

```
bin/webrick/check_progress
```

...and check the percentage of nodes to have completed step 2.

Site status after running step 2:

After every agent node has checked in once:

- CA will create dangerous certs? **NO.** (fixed!)
- Agents can be spoofed by agent certs? **NO.** (fixed!)
- Potentially dangerous certs are still valid? **YES.** (not fixed)
- Agents can operate normally and receive catalogs from master? **YES.** (business as usual)

Your site is now protected. However, all of the master's previous DNS names are unsafe to use for the remaining lifetime of the CA. If you are content to leave the puppet master on the new DNS name, you can stop now and change your new node bootstrapping process to use the new name; otherwise, continue to step 3.

Schedule steps 3-5 carefully, as step 4 entails a temporary disruption of service.

You should not run step 3 until all agents have run once and your full site is protected.

Step 3

This step:

- Generates a new CA certificate.
- Configures Puppet to sign any NEW certificate requests using the new CA.
- Prepares the puppet master to trust agent certificates from both the new and the old CA.

This step modifies only the puppet master. You can perform the next step immediately.

PE Users

From the top directory of this module, run the following:

```
bin/pe_step3_generate_new_authority
```

Other Users

Stop the WEBrick puppet master you started in step 2. From the top directory of this module, run the following:

```
bin/webrick/webrick_step3_generate_new_authority
```

Do not re-start the puppet master; proceed directly to step 4.

Site status after running step 3:

- CA will create dangerous certs? **NO.** (fixed!)
- Agents can be spoofed by agent certs? **NO.** (fixed!)
- Potentially dangerous certs are still valid? **YES.** (not fixed)
- Agents can operate normally and receive catalogs from master? **YES.** (business as usual)

Step 4

This step:

- Adds the `cve20113972::step4` class to all agent catalogs. This class:
  - Moves the agent's `ssldir` to a backup location.
  - Securely configures the agent to trust the new CA (and *only* the new CA).
  - Configures the agent to contact the master at its old DNS name
  - Restarts puppet agent.
  - Submits a new certificate signing request.

This step modifies agent nodes. **All agent nodes must run once before performing the next step.**

PE Users

From the top directory of this module, run the following:

```
bin/pe_step4_migrate_agents_to_new_authority
```

Other Users

From the top directory of this module, run the following:

```
bin/webrick/webrick_step4_migrate_agents_to_new_authority
```

Start a WEBrick puppet master server as described in step 1.

Checking agent status

This step is not complete until every agent node has run once. You can use the included `check_progress` script to view a summary of your nodes' progress through this step; simply run:

```
bin/check_progress
```

or:

```
bin/webrick/check_progress
```

...and check the percentage of nodes to have completed step 4.

Site status after running step 4:

After every agent node has checked in once:

- CA will create dangerous certs? **NO.** (fixed!)
- Agents can be spoofed by agent certs? **NO.** (fixed!)
- Potentially dangerous certs are still valid? **NO.** (fixed!)
- Agents can operate normally and receive catalogs from master? **NO.** (disruption of service)

The puppet master's previous DNS names have been rehabilitated, and are safe to use in the future; accordingly, this step restores the agent's configuration so it will use the previous name.

Note that agent nodes which have run a step 4 catalog will be unable to retrieve and run their normal catalogs until the end of step 5.

You should not run step 5 until all agents have run once. If you run step 5 too early, any agents who have not run their step 4 catalogs **will be in an "orphaned" state** and must be repaired manually. Use the `check_progress` script to check how much of your population has been migrated to the new CA.

Orphaned nodes can be repaired by logging in, moving the `ssldir` to a new location, and restarting puppet agent. Run `puppet agent --configprint ssldir` to locate an agent node's `ssldir`.

Step 5

This step:

- Cleans out the puppet master's certificate from the old CA.
- Issues the master a certificate signed by the new CA.
- Configures the master to distrust agent certificates from the old CA.

This step modifies only the puppet master.

PE Users

From the top directory of this module, run the following:

```
bin/pe_step5_migrate_the_master
```

Other Users

Stop the WEBrick puppet master you started in step 4. From the top directory of this module, run the following:

```
bin/webrick/webrick_step5_migrate_the_master
```

Re-start **your original production-scale puppet master server.**

Site status after running step 5:

After every agent node has checked in once:

- CA will create dangerous certs? **NO.** (fixed!)
- Agents can be spoofed by agent certs? **NO.** (fixed!)
- Potentially dangerous certs are still valid? **NO.** (fixed!)
- Agents can operate normally and receive catalogs from master? **YES.** (service has resumed)

This step completes the full remediation.

You must still sign any pending certificate signing requests to reenable normal agent traffic -- if you don't use autosign, you should manually sign these requests with the `puppet cert --list` and `puppet cert --sign` commands. (If you recognize every certname in the list of CSRs and are confident that none were submitted by malicious nodes, you may wish to use `puppet cert --sign --all`.)