A tool for deploying and detecting use of Active Directory honeytokens https://www.secureworks.com/blog/dcept

agent	Initial commit	a year ago
server	Fixed ConfigReader realm check	a year ago
AUTHORS	Initial commit	a year ago
LICENSE	Initial commit	a year ago
README.md	Update README.md	8 months ago

README.md

DCEPT

DCEPT (Domain Controller Enticing Password Tripwire) is a honeytoken-based tripwire for Microsoft's Active Directory. Honeytokens are pieces of information intentionally littered on system so they can be discovered by an intruder. In the case of DCEPT, the honeytokens are credentials that would only be known by a someone extracting them from memory. A logon attempt using these faux credentials would mean someone was inside the network and is attempting privilege escalation to domain administrator.

This proof of concept is being released as open source to benefit Windows system administrators. The goal of this project was to provide a free, simple, honeytoken deployment tool as well as educate administrators about the nature of these attacks. We encourage contributors to build on what we have done and welcome feedback. Has DCEPT helped your organization spot an intrusion before it was too late? We would like to hear from you.

More information about this research project can be found here: https://www.secureworks.com/blog/dcept

Overview

There are three components to DCEPT. The first is an agent written in C# that caches honeytokens in memory on the endpoints. The tokens themselves are invalid credentials and pose no risk of compromise. Honeytokens are requested at regular intervals and are uniquely associated with a workstation for a particular window of time; therefore providing a forensic timeline. In the event a honeytoken is used on a different workstation at a later date, its point of origin is still known, potentially narrowing the scope of an investigation.

The second is a server component that generates and issues honeytokens to requesting endpoints. Generated tokens are stored in a database along with the timestamp and endpoint that requested it.

A third component acts as a monitor that passively listens for logon attempts. In order to capture the necessary packets, the DCEPT interface needs to be on the same network as the domain controller.

Getting Started

A Docker container build for the server components is provided, making deployment a simple process. Before you can use DCEPT, you must have Docker installed on your system. Consult the Docker website for [installation instructions] (https://docs.docker.com/engine/installation/).

Configuration

The configuration file is named “dcept.cfg” and must be modified before running the Docker container. Currently only notifications via rsyslog are supported. Configure syslog_host to point to your SIEM's syslog server.

Multi-server Architecture

DCEPT can be run in standalone or a multi-server configuration. By default, DCEPT runs as a master node where it is responsible for generating credentials and sniffing out the authentication requests. Traffic from multiple DCs can be monitored by a single DCEPT instance using network taps. Alternatively, DCEPT can run as a slave node by setting the “master_node” option to point to a master node. In this configuration, the slave nodes sniff and then relay relevant data to the master node where it is matched against the database of credentials.

Building the Docker Image

```
root@host:~# cd server
root@host:~# ./docker_build.sh
```

Running the Docker Image as a Container

Run the Docker container interactively with the following command:

```
root@host:~# cd server
root@host:~# ./launcher.sh
```

Run the container in the background with the following command:

```
root@host:~# cd server
root@host:~# ./daemon.sh
```

Building the Agent

The agent is provided as C# source code only, designed to be audited and compiled by the network administrator before deploying to endpoints. If you are compiling on a Windows system, Microsoft provides Visual Studio Express for free which can be downloaded [here] (https://www.visualstudio.com/products/visual-studio-express-vs).

Configuring the Agent

The agent configuration is hardcoded and must be altered prior to compilation. Toward the top of the code, you will find two constants *URL* and *PARAM*. The URL should point to the DCEPT Generation Server. The URL can also contain any number of arbitrary directories/subdirectories. This is simply cosmetic and intended to intrigue a hacker should they come across the URL. The *PARAM* constant is how the agent passes the endpoint hostname to the Generation server. The parameter name can also be changed for cosmetic purposes, but be sure that is reflected in the generation server configuration file.

IMPORTANT: Using names such as “honeytoken” or anything else that might suggest you are using DCEPT to a hacker is both counterintuitive and highly discouraged.

```
// Edit this to point to your Honeytoken server URL
static string URL="http://not-a-dcept-server-wink.domain.lan/backup/auth/nonsense  ";
static string PARAM="machine";
```

Compiling on Ubuntu

If you prefer to compile from an Ubuntu system, you can use mono. If you don't already have it installed, you can run the following command to install the mono development packages and C# compiler.

```
root@host:~# apt-get install monodevelop mono-mcs
```

Once mono is installed, change your working directory and then run the following to compile the source code.

```
root@host:~# mcs ht-agent.cs -r:System.Data.dll -r:System.Web.Extensions.dll -r:System.Web.Services
```

Deploying the Agent

How the agent is deployed will vary from organization to organization and is entirely up to you. Deployment in a way that would leave *valid* domain administrator credentials cached on the endpoints (e.g. psexec) is highly discouraged.

Testing

Run the following command to get an interactive shell inside the container.

```
root@host:~# docker exec -it dcept /bin/bash
```

tcpreplay is installed inside the docker container along with a sample pcap for testing purposes. While DCEPT is running, execute the following from within the container:

```
root@host:~# tcpreplay -i <interface> /opt/dcept/example.pcap
```