OSMOSIS Demo

Some assumptions:

We are covering default installations, no changes or plugins

Non of the insecure-by-architecture “features” are tackled

No “fake” environments (remember those demos with a very, very specific set up you maybe see only once – in the researcher’s lab?)

Cacti

Bugs:

Cross Site Request Forgery (CSRF)
Exec-like functions called on untrusted input

Cacti

Fill in shell-style command as username in "Graph Export" form

Gets executed on Cacti server when admin clicks on link

.. or visits malicious website

Cacti

Sending a targeted attack URL

Pro:

Con:

Need to know Cacti installation URL

Admin needs to access link or site with link to trigger exploit

Outgoing connection may be restricted

Admin needs to be logged in

(Not really..)

check_mk

Bugs:

Cross Site Request Forgery (CSRF)

Exec-like functions called on untrusted input

Cross Site Scripting (XSS)

check_mk

Exploits a "feature" in the wato plugin (snapshots)

A specially crafted snapshot is uploaded and unpacked

This can be achieved due to missing CSRF tokens

Unpacked == Executed, as snapshots contain plain python code

check_mk

Sending a targeted attack URL

Pro:

Persistent Shell!

Con:

Need to know check_mk installation URL

Admin needs to access link or site with link to trigger exploit

Admin needs to be logged in

Outgoing connection may be restricted

check_mk

Need to know the application URL? We can do better..

Let's combine a XSS with the snapshot "feature"

What get's definitely displayed? The agent string!

And we can re-use the connection. Who needs firewalls?

check_mk

Modify a check_mk agent

Pro:

Persistent Shell!

URL not needed for this attack as it is "in browser"

Using existing connection for outgoing shell

Is only triggered when Admin logs in

Con:

Need (root) access to a monitored system

check_mk

Owning a target system is not always an option (some are actually hardened!)

We can own - with just one simple ssh _attempt_

A XSS triggers a CSRF triggers an upload that triggers a shell

We need logwatch for that. Default installation will do.

XSS gets triggered by a specially crafted ssh connection string / username containing JS

For the old fart^H^H^H^H hackers in this room ->

Interlude: Terminal Escape Character Issues

Anyone remember THIS?

[prev in list] [next in list] [prev in thread] [next in thread]

List: bugtraq
Subject: Terminal Emulator Security Issues
From: H D Moore <termulation () digitaloffense ! net>
Date: 2003-02-24 21:02:52
[Download message RAW]

%s/terminal emulator/browser/g

check_mk

Use one SSH connection string

(a.k.a one ssh connect _attempt_ to own you..)

Pro:

Ultra stealth

No need to know URL beforehand

Is only triggered when Admin logs in

Con:

Outgoing Connection need to be possible (DNS/ICMP shells anyone?)

check_mk

Here's one for the evil minded:

Spit out that string 0.0.0.0/0 on Port 22

Set up ev1l web Site

Wait for them shells popping

Profit!