Welcome to Rootkit Country

Graeme Neilson

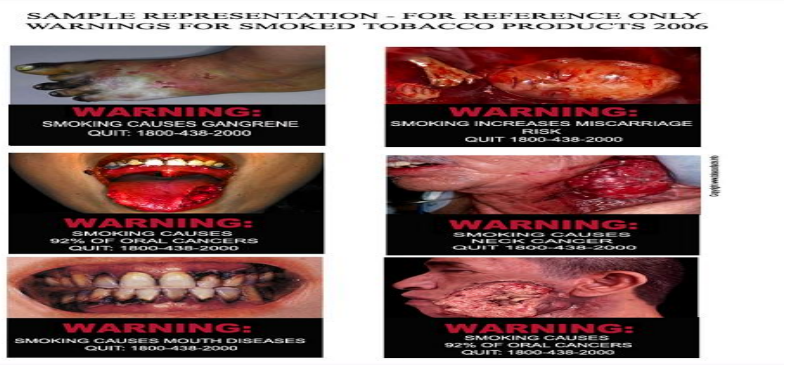
Security Consultant & Researcher

Aura Software Security

graeme@aurasoftwaresecurity.co.nz

Rootkit == cancerous software

"*A rootkit is software that enables continued privileged access to a computer while actively hiding its presence from administrators by subverting standard operating system functionality or other applications.*"
Wikipedia



BIOS Kernel System Applications

Patches and Gum

- File integrity checks (checksums, hashes)
- Immutable files (secure run levels, read only filesystems)
- Mandatory access control
- Memory access restrictions
- Signed software
- Encrypted software

UTMs / Firewalls / Routers?

Why are they a target?
Re-route traffic, mirror traffic, layer 2 control, VPN endpoints, management network connectivity, choke points for many networks.

As endpoints physical access can be out of the owner's control.

How are they attacked?
Social Engineering, Insider, Physical Access, Supply Chain, [Exploits]

[Often remote attack surface for exploits is very limited or non existent.
Not going to discuss exploits / vulnerabilities.]

Can I trust the integrity of the operating systems that run these devices?

How easy is it to rootkit these devices?

Platforms


NETGEAR®
Connect with Innovation™


NOKIA
Connecting People


ERICSSON


Check Point®
SOFTWARE TECHNOLOGIES LTD.


WatchGuard™


CISCO


McAfee


FORTINET®


juniper
NETWORKS®


SONICWALL

Roll your 0wn

1. **Go shopping.**

2. **Obtain firmware.** Downloadable, backup, compact flash, hard disk, VM

3. **Identify the firmware.** Linux, FreeBSD, vxWorks, proprietary (some RTOS)

4. **Gain root level access.** Not like developing on a desktop OS where we have root by default. May need to reverse engineer firmware / package formats, break restricted shells, crack passwords, crack / bypass encryption etc

5. **Determine layer to attack.** BIOS, Kernel, System, Application

6. **Installation method.** Package, OS Update, Physical

7. **Welcome to Rootkit Country**

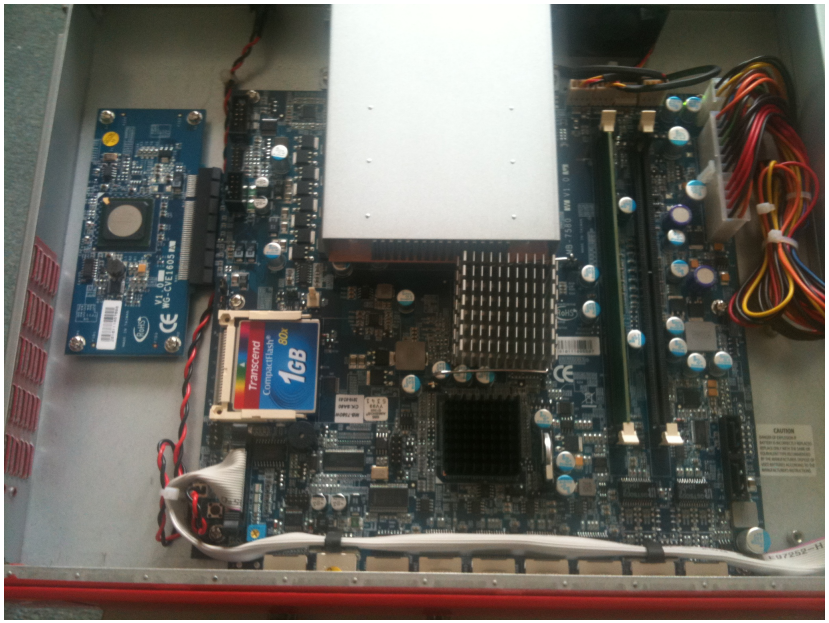
WatchGuard

OS	XTMOS Linux 2.6.21
Arch	i686
Bootloader	GRUB
Storage	Removable CF
Firmware Format	Gzip image with custom header
Restricted Shell	yes
Root access	Hardcoded password
Memory	-
Integrity	None

SilkGuard Rootkit

Root access:
- remount rootfs read write
- add static compiled shell busybox
- add authorized_key to /root/.ssh/
- SSH on port 4118

Layers to attack:
- load our own kernel / libraries / applications
- loadable kernel modules
- replace



Netgear ProSecure

OS	Linux 2.6.21
Arch	MIPS
Bootloader	GRUB
Storage	Removable CF
Firmware Format	SquashFS
Restricted Shell	no
Root access	Random password at boot
File System	RO unionfs
Memory	/dev/mem and /dev/kmem readable
Integrity	none

NetHill Rootkit

Root access:
• use squashsfs 3.4 (big-endian support)
• write new rootfs.img with root password removed

Layers to attack:
• From a Linux Debian MIPS VM we can add a missing apt-get library, uncomment sources and install compiler, debugger etc

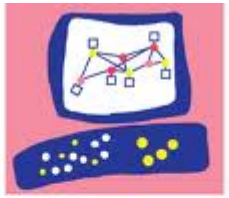
• system-map & config present on system

• LKM rootkit, system or application

• Replace

CheckPoint Secure Platform

OS	CP Linux (RHEL) 2.6.18
Arch	i686 / Virtual
Bootloader	GRUB
Storage	ISO
Firmware Format	ISO
Restricted Shell	Yes
Root access	Yes
File System	ext
Memory	Restricted access
Integrity	none



Check Point®
SOFTWARE TECHNOLOGIES LTD.

LuckyPoint Rootkit

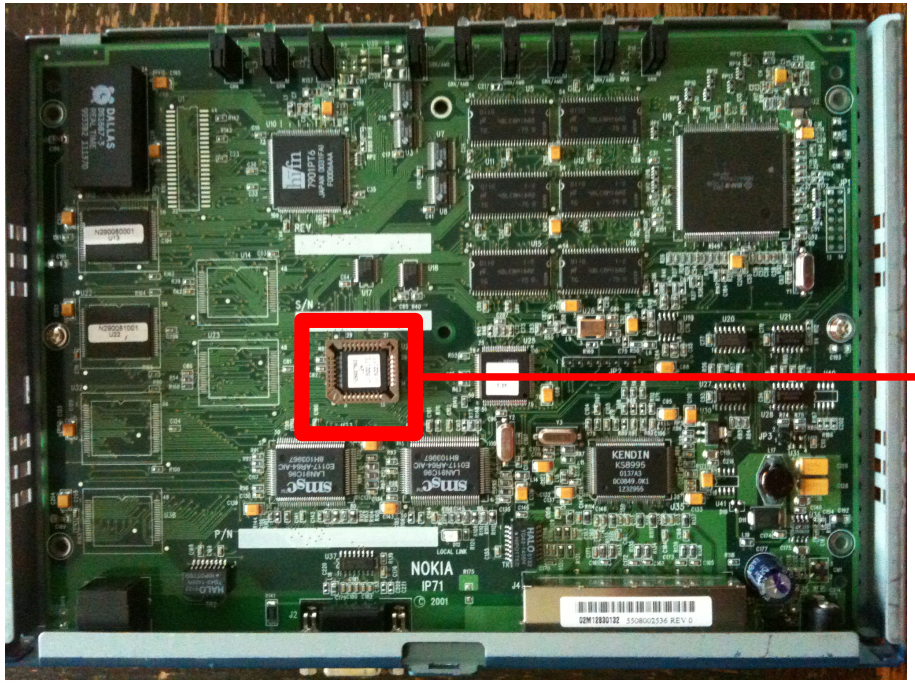
Root access:
Built in through “expert” mode

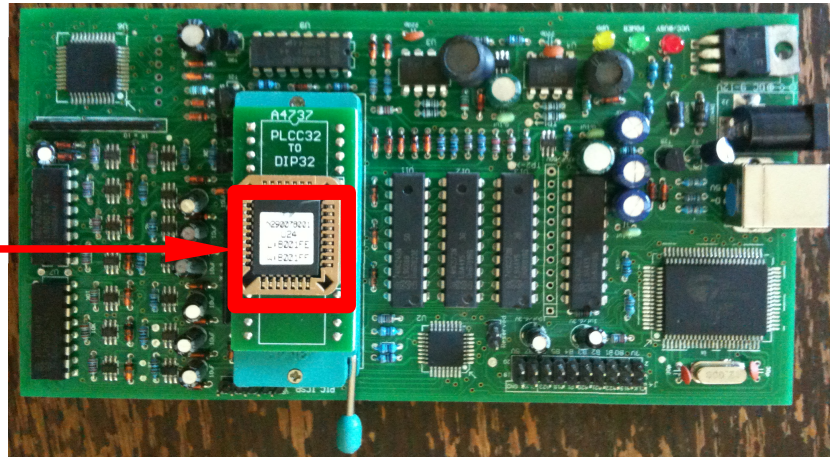
Layers to attack:

- System map and config available but
- No /dev/kmem and /dev/mem restricted to first 2056 records
- Libraries and applications
- Linux with some custom packages and scripts.

Checkpoint Nokia

Nokia IP71 common endpoint device for CheckPoint Secure Platform
- has removable, flashable BIOS
- BIOS integrity check is a simple checksum
- BIOS modification and rootkit possible





Fortinet FortiOS

OS	FortiOS Linux
Arch	i686
Bootloader	GRUB
Storage	Removable CF
Firmware Format	Gzip
Restricted Shell	yes
Root access	no
File System	Encrypted AES CBC
Memory	
Integrity	FortiBIOS, firmware encrypted, signed & hashed

Export-F Rootkit

Root access:
Encryption used has weaknesses as watermarks reveal a disk image format and the location of MBR, kernel and root file system can be determined
Fortigate will load firmware even if there is no certificate, no hash and is unencrypted.
Verification of firmware is:

- Start of MBR must contain a filename matching a device & version ID
- Kernel must be called “fortikernel.out”

Layers to attack:
Lload replacement kernel and file system

Sonicwall

OS	SonicOS vxWorks
Arch	i686
Bootloader	?
Storage	Secure Compact Flash
Firmware Format	Encrypted / Compressed
Restricted Shell	Yes
Root access	No
File System	vxWorks
Memory	Restricted access
Integrity	Encryption

Cancer Free

Root access:

- Removable Storage Compact Flash
  but its unreadable...
- Removable BIOS
  but its unreadable...
- Firmware can be backed up
  but its signed...

Cisco IOS - Da Los Rootkit

OS	IOS
Arch	MIPS
Bootloader	Proprietary
Storage	Flash
Firmware Format	Compressed
Restricted Shell	Yes
Root access	No
File System	Memory
Memory	
Integrity	Checksum

Juniper ScreenOS

OS	ScreenOS
Arch	PowerPC
Bootloader	Proprietary
Storage	Flash
Firmware Format	Compressed (custom LZMA or GZIP)
Restricted Shell	Yes
Root access	No
File System	Memory
Memory	Flat memory model
Integrity	Checksum, optional signature

Junboro Light Rootkit

Root Access:

- Firmware is compressed (non standard LZMA header)
- Reverse engineer format and then proprietary OS (IDA) to find useful functionality to subvert.
- Firmware checksum algorithm can be reverse engineered
- Firmware is signed but certificate can be loaded and unloaded from device by root

Layers to attack:

- Flat memory, monolithic firmware so access to everything
- Hand code PowerPC ASM into firmware to backdoor login, subvert certificate check (in boot loader) and provide new functionality

Juniper JUNOS

OS	ScreenOS
Arch	i686 / Virtual
Bootloader	FreeBSD
Storage	Flash, HDD
Firmware Format	Package
Restricted Shell	Yes
Root access	Yes
File System	RO iso9660
Memory	Restricted access
Integrity	Veriexec, secure level 1, hash, optional signature

Junboro Rootkit

Root access

- Have root by default but there are restrictions:
- JUNOS binaries are symlinks from rw fs to iso9660 ro fs
- Secure run level 1 is set
- Veriexec used for integrity and to stop unknown binaries running

Layers to attack:

- +x shell scripts will not run directly but **will run** if invoked by /bin/sh
- JUNOS doesn't require/enforce signed packages
- Install trojaned package using +INSTALL script which
  - turn offs veriexec (rm /etc/rc symlink, cp new /etc/rc with veriexec off)
  - installs trojaned binaries / libraries / backdoor
  - forces pkg requires reboot flag on

Demos

Make	Arch	OS
Fortinet	Intel	Linux
Juniper	PPC	ScreenOS
Juniper	VM	JUNOS

Conclusion

- Many “secure” UTM operating systems are modified open source operating systems
- Few have robust integrity checking
- Many defenses can be bypassed as they are implemented weakly
- Do periodic offline hash comparisons of system binaries
- Validate your supply chain
- The best protection mechanisms are:
  - Encrypt firmware
  - Enforce signed firmware/packages

References

Runtime Kernel Mem Patching,
http://vxheavens.com/lib/vsc07.html, Silvio Cesare

Killing the myth of Cisco IOS rootkits: DIK (Da Ios rootKit),
http://eusecwest.com/esw08/esw08-muniz.pdf

Hacking Grub for fun and profit,
Phrack Volume 0x0b, Issue 0x3f, CoolQ

Static Kernel Patching,
Phrack Volume 0x0b, Issue 0x3c, jbtzhm

Playing Games With Kernel Memory ... FreeBSD Style,
Phrack Volume 0x0b, Issue 0x3f, Joseph Kong

Implementing and detecting ACPI BIOS rootkit,
http://www.blackhat.com/presentations/bh-federal-06/ BH-Fed-06-Heasman.pdf

Questions?