Let's Screw With nMap

NOSEY LITTLE BASTARD
AREN'T YOU?

DefCon 21, Las Vegas 2013

Hellfire Security

Gregory Pickett, CISSP, GCIA, GPEN
Chicago, Illinois

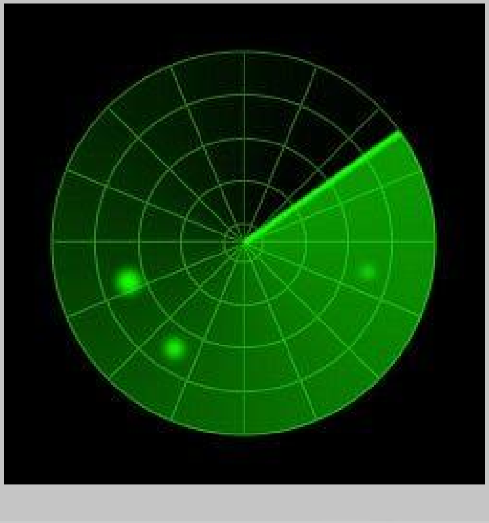
gregory.pickett@hellfiresecurity.com

Overview

- Nosey Bastards!
- All About Packet Normalization
- Working It All Out
- Putting It Into Practice
- Finishing Up

Network Defenders

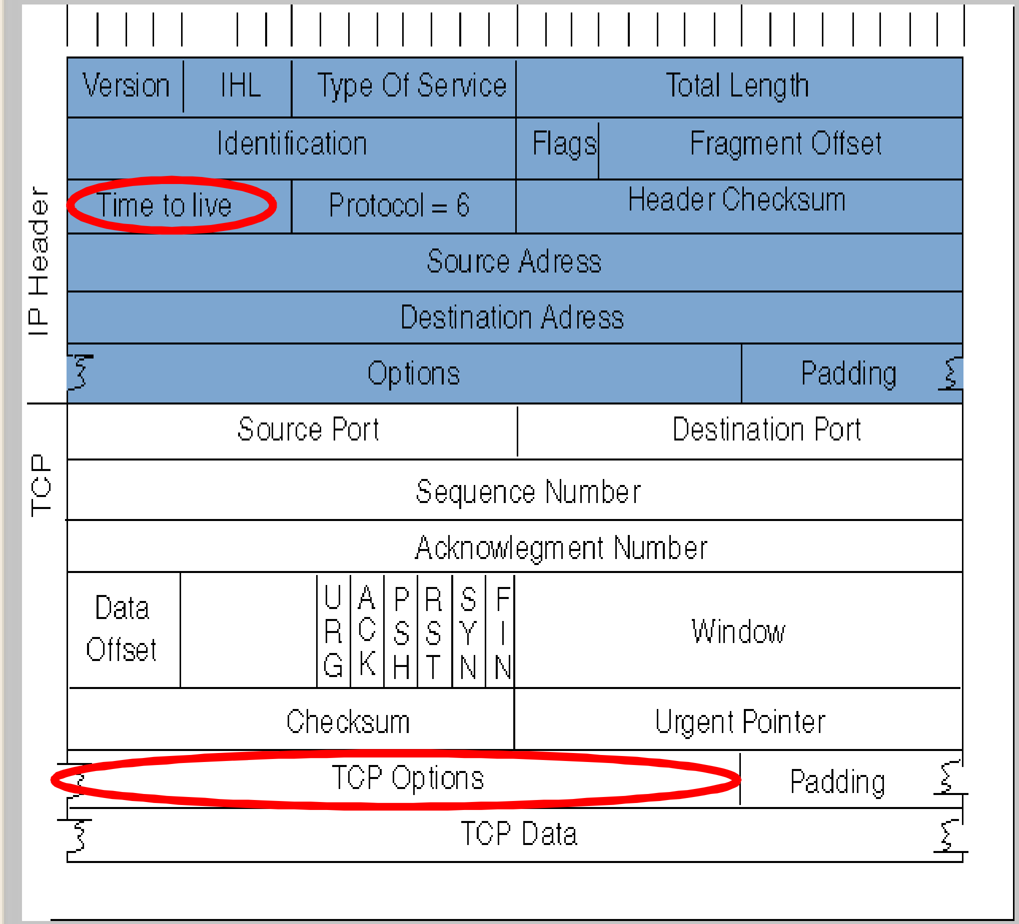
- We see scans and probes of our network every day
- From the inside and from the outside
- Everybody is targeting us
- Identifying our assets



How They Do It

- Network stack implementation is highly discretionary
- Differences identify the operating system type and version
- Allowing Attackers to identify their targets
- By matching the headers of their target to known operating system implementations

If your target . . .



- **Has a TTL of 128**
- **Uses the following options**
  - **MSS of 1460**
  - **Single NOP**
  - **Window Size 0**
  - **Single NOP**
  - **Single NOP**
  - **Ending SACK**

. . . then it's likely a Windows 2003 Sever!

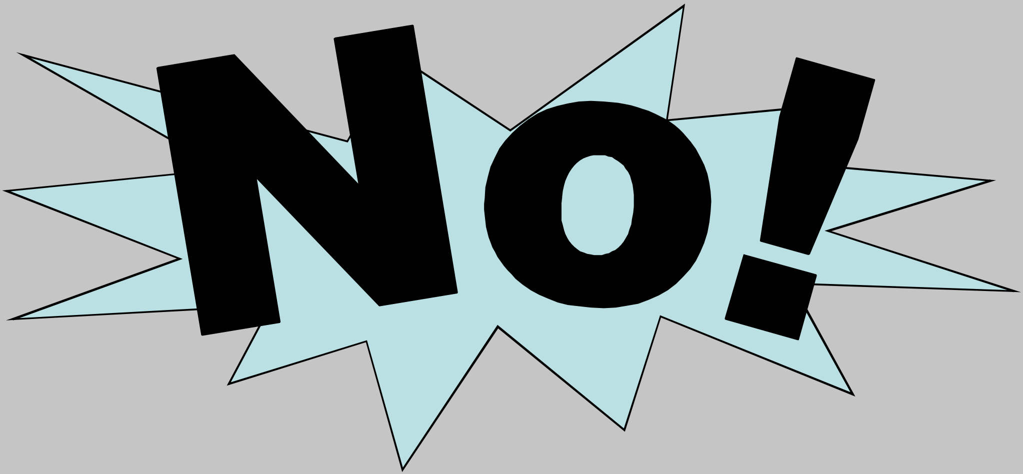
Implications

- If they identify your assets …
- They know their weaknesses
- How to attack them successfully
- Without triggering your sensors

TSA-Style patdowns . . .

It's fact of life

But does it have to be?



No!

Why can’t we . . .

- Remove the differences
- To remove their advantage
- Strip them of their ability to fingerprint
- To significantly reduce their chance of success

My Answer

Packet

HELLO
MY NAME IS
Normal

ization

OK. What is packet normalization?

- Not an entirely developed concept
- Many expressions but most incomplete …

Normalization vs. Scrubbing

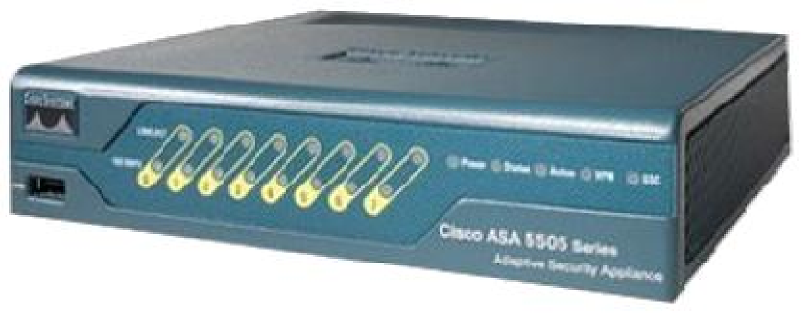
- **Scrubbing** is to do away with; cancel
- **Normalization** is to make normal, especially to cause to conform to a standard or norm
- Both are seen in varying degrees

Scrubbing

- Used by a number of firewalls
  - Randomize IP ID
  - Clear IP DF
- Also …
  - Set IP tos/dscp, and ttl
  - IP Fragment Reassembly
- Primarily Concern
  - Policy Violations
  - Abnormal Packets
  - Abnormal Flows

Scrubbing

- Used by some network devices such as Cisco ACE and ASA
  - Random TCP SEQ
  - Clear TCP Reserved, and URG
  - Clears TCP Options
  - Minimum IP TTL
- Fragment Reassembly too …
- Primarily Concern
  - Policy Violations
  - Abnormal Packets
  - Abnormal Flows



Incoming Normalization

- Used by IPS and IDS devices
  - IP Fragment Reassembly
  - IP TTL Evasion
- Primarily Concern
  - Detect Attacks
  - Detection Evasion

Masquerading

- Examples
  - IP Personality
  - Morph
  - IP Morph
- Pretends to be …
- Modifies the stack
- Host Only

IP Personality

SYN ACK LABS

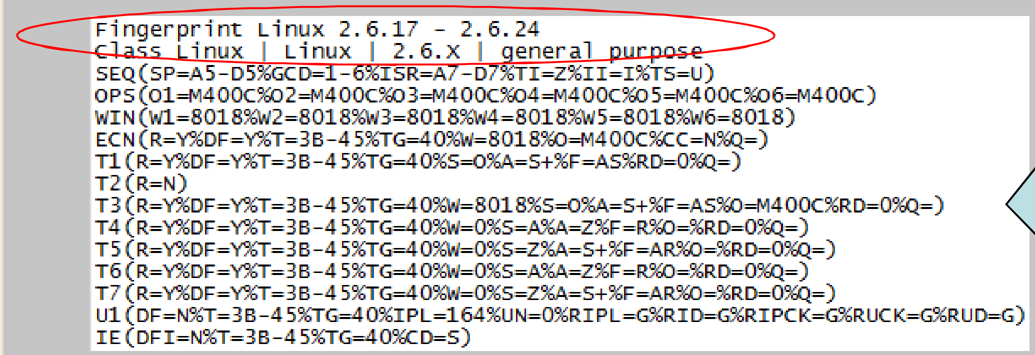
IpMorph

Outgoing Normalization?

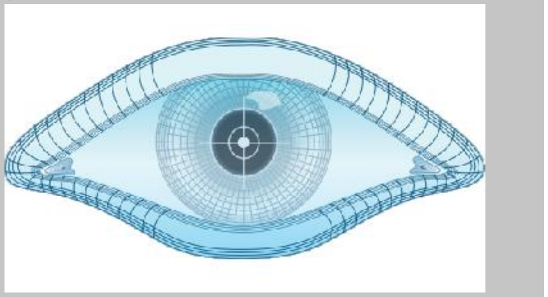
Not Really

Fingerprinting Process

- TCP, UDP, and ICMP probes are sent
- Compile results into fingerprint



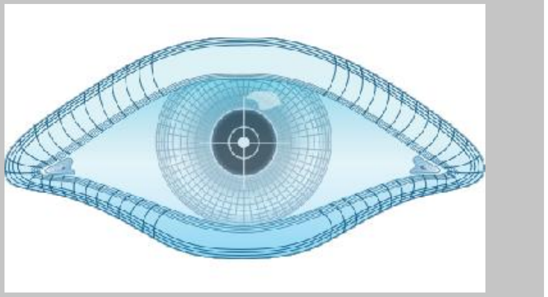
```
Fingerprint Linux 2.6.17 - 2.6.24
Class Linux | Linux | 2.6.x | general purpose
SEQ(SP=A5-D5%GCD=1-6%ISR=A7-D7%TI=Z%II=I%TS=U)
OPS(O1=M400C%O2=M400C%O3=M400C%O4=M400C%O5=M400C%O6=M400C)
WIN(W1=8018%W2=8018%W3=8018%W4=8018%W5=8018%W6=8018)
ECN(R=Y%DF=Y%T=3B-45%TG=40%W=8018%O=M400C%CC=N%Q=)
T1(R=Y%DF=Y%T=3B-45%TG=40%S=O%A=S+%F=AS%RD=0%Q=)
T2(R=N)
T3(R=Y%DF=Y%T=3B-45%TG=40%W=8018%S=O%A=S+%F=AS%O=M400C%RD=0%Q=)
T4(R=Y%DF=Y%T=3B-45%TG=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)
T5(R=Y%DF=Y%T=3B-45%TG=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)
T6(R=Y%DF=Y%T=3B-45%TG=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)
T7(R=Y%DF=Y%T=3B-45%TG=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)
U1(DF=N%T=3B-45%TG=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)
IE(DFI=N%T=3B-45%TG=40%CD=S)
```



- Compare against database
- Identify operating system

Where to Start?

- Nmap fingerprint database
- What about other fingerprinting tools?
  - xprobe2
  - SinFP
  - Vulnerability scanners … Nessus, Others
- Best to disrupt any existing patterns



Scrubbing

- Clear out any unnecessary values
  - IP ToS/DSCP/Traffic Class Cleared
  - IP ECN Cleared
  - TCP URG Flag and URG Pointer Cleared
- Randomize anything that you can
  - IP ID
- IP TTL/HOP Limit? TCP Options?

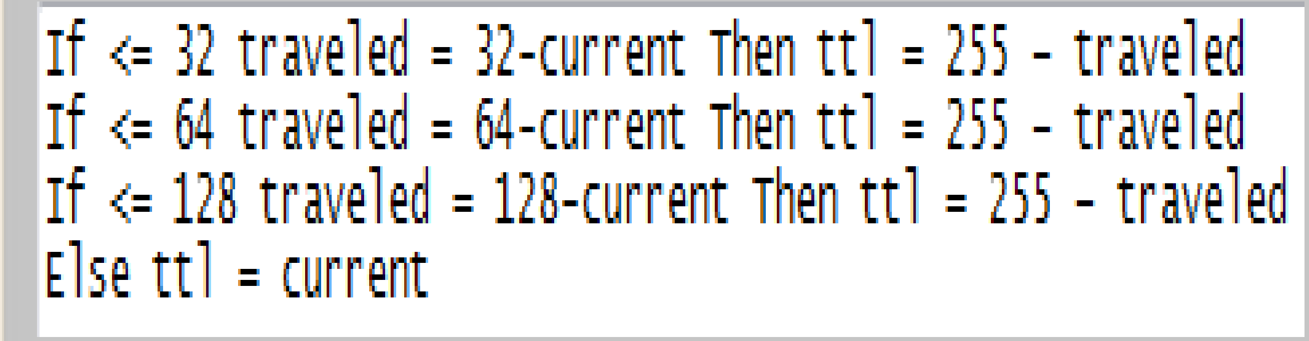
OuRgoket Normalization

To The Rescue!

Normalizing
(IP Time-To-Live / Hop Limit)

- Make some assumptions
  - Originally Well-Known TTL
  - Decrements Only
  - Traveled < 32 hops
- Back into Original Starting TTL
- Estimate number of hops traveled
- Recalibrate current TTL
- Using Starting TTL of 255

Normalizing
(IP Time-To-Live / Hop Limit)



```
If <= 32 traveled = 32-current Then ttl = 255 - traveled
If <= 64 traveled = 64-current Then ttl = 255 - traveled
If <= 128 traveled = 128-current Then ttl = 255 - traveled
Else ttl = current
```

- Start with the lowest well known TTL first!
- Several exceptions to this normalization …
- Will be discussed later

Normalizing (TCP Options)

- Assumptions
  - Only Few Well Known Options Needed
  - Order is unimportant
- Requirement …Values can't be changed
- Read necessary options
- Discard the rest
- Rewrite options in proper order
- NOP … till the end of the options

Normalizing (TCP Options)

- Options selected … And their order
  - MSS
  - Window
  - SACK
  - MD5 … if present
- After processing …

MSS = 1460
Window = 0
SACK
NOP
NOP
NOP

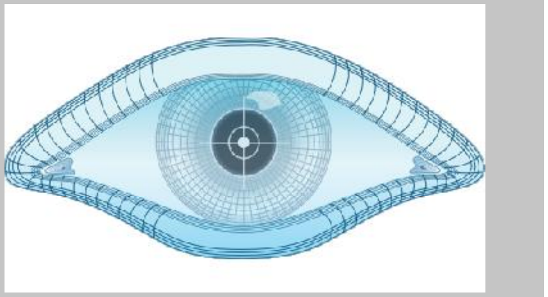
Making everyone look the same

Putting It All Together

With IDGuard

Selecting The Platform

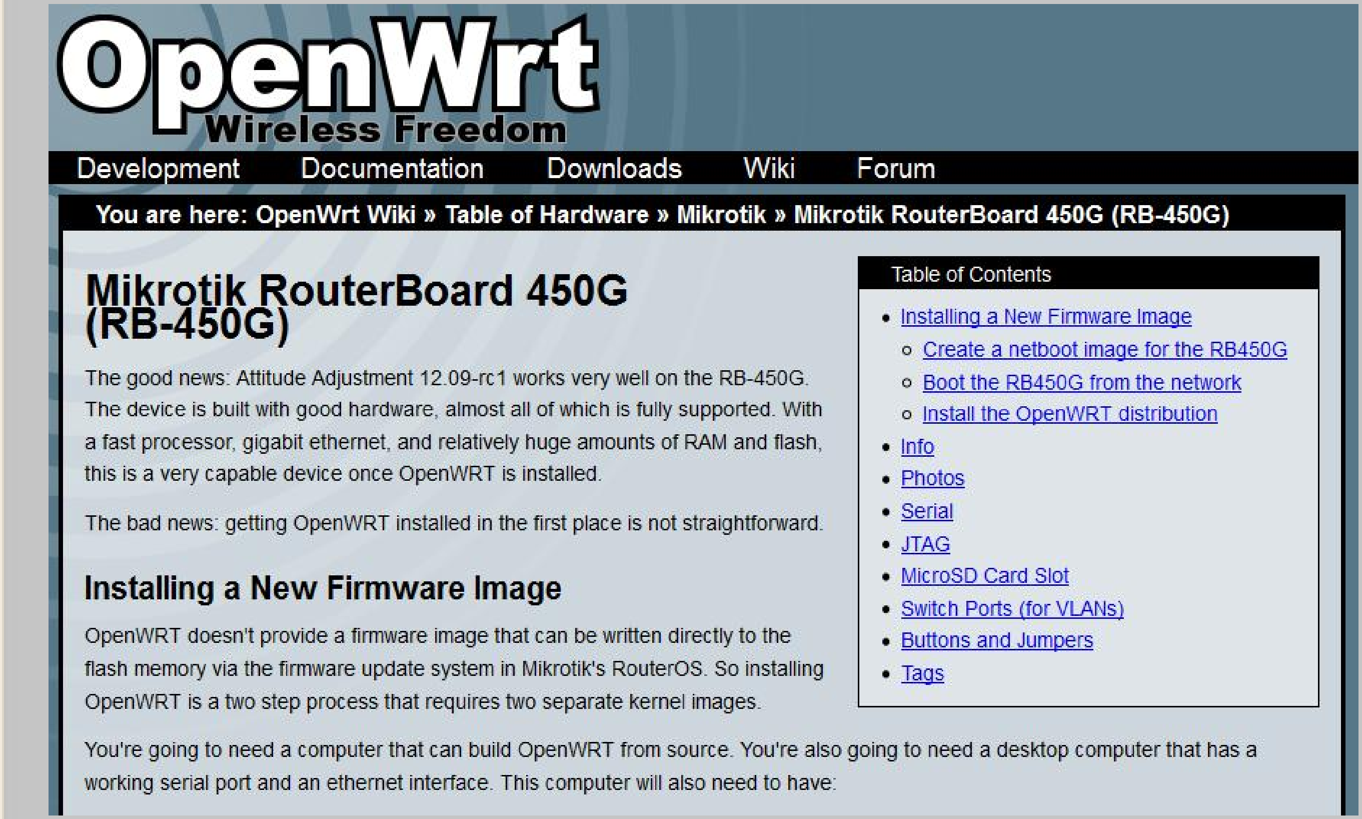
- Identified Suitable Hardware
  - Already Modified By Others
  - Documentation Available … Mikrotik Routerboards
- Identified Suitable Operating System
  - Available Base
  - Writeable File System …OpenWrt



Deploying to Hardware

- Purchase the hardware from a local vendor
- Create a netboot image for the RB450G
- Setup dhcp & tftp netboot environment
- Connect to the routerboard
- Configure routerboard for DHCP
- Netboot routerboard and flash
- Load kernel module manually or with a package
- Configure Firewall

Deploying to Hardware



You are here: OpenWrt Wiki » Table of Hardware » Mikrotik » Mikrotik RouterBoard 450G (RB-450G)

Mikrotik RouterBoard 450G (RB-450G)

The good news: Attitude Adjustment 12.09-rc1 works very well on the RB-450G. The device is built with good hardware, almost all of which is fully supported. With a fast processor, gigabit ethernet, and relatively huge amounts of RAM and flash, this is a very capable device once OpenWRT is installed.

The bad news: getting OpenWRT installed in the first place is not straightforward.

Table of Contents

Installing a New Firmware Image

OpenWRT doesn't provide a firmware image that can be written directly to the flash memory via the firmware update system in Mikrotik's RouterOS. So installing OpenWRT is a two step process that requires two separate kernel images.

You're going to need a computer that can build OpenWRT from source. You're also going to need a desktop computer that has a working serial port and an ethernet interface. This computer will also need to have:

OK . . . What worked?

I am really tired of those nosey bastards!

What Didn’t Work

- ToS/DSCP/Traffic Class Clearing
- ECN Clearing
- URG Flag and URG Pointer Clearing
- IP ID Randomization
- DF Clearing

. . . the Scrubbing

What Worked

- TTL Standardizing
- TCP Option Standardizing

. . . the Normalization

End Results

Operating System	Unprotected	Protected
Windows 7	Microsoft Windows 7 | 2008	Allied Telesyn AlliedWare
Windows Server 2003	Microsoft Windows 2003	Allied Telesyn AlliedWare
Ubuntu Desktop 11.10	Linux 2.6.X | 3.X	Cisco IOS 12.X
Red Hat Enterprise Linux 6	Linux 2.6.X | 3.X	D-Link embedded

Other Effects

- Nmap
  - Network Distance
- Other Fingerprinting
  - Xprobe2
  - SinFP
  - Nessus …
- Other Tools
  - ping
  - traceroute

Demonstration

Challenges

- Authorized Activity
- Other Methods
  - Banners and Direct Query
  - Identification Through Layer-7

Challenges

- Authorized Activity
  - Scanners
  - Management Platforms
- Resolution
  - IDGuard Excludes Them …

Challenges

- Banners and Direct Query
  - Windows Networking Available
  - Application-Layer Query
  - OS Details in Reply
- Resolution
  - Perimeter Network
  - Internal Network

Concerns

- Connectivity
  - Fragmentation
    - Upstream
    - Downstream
  - TTL Attenuation
  - TTL Special Uses
- TCP Options Sensitivity?
- Link-Local Routing Protocols

Concern

- Upstream Fragmentation
  - IP ID Randomized
  - "Fragmentation Needed" ICMP Message Received
  - Host is confused
  - Keeps sending original packet
- Resolution
  - IDGuard Clears DF

Concern

- Downstream Fragmentation
  - Each fragment given a different IP ID
  - Destination can’t reassemble original
- Resolution
  - Access switch placement
  - IDGuard Excludes Fragments

Concern

- TTL Attenuation
  - Packet travels more than 32 hops
  - Not all these hops are accounted for …
  - Packet TTL is continually extended
  - Routing Loop occurs
- Resolution
  - Access Switch Placement

Concern

- TTL Special Uses
  - TTL recalibrated
  - TTL never runs out
  - No Intermediate hop reports
  - Traceroute fails
- Resolution
  - IDGuard Excludes ICMP Echo Requests
  - IDGuard Excludes the UDP traceroute range

Concern

- Link-Local Routing Protocols
  - RIP packets have a TTL of 1
  - TTL of 255 is abnormal
  - Packet is malformed
- Resolution
  - IDGuard Excludes Routing Protocols

Concerns

- Performance
- Break Something
  - Poorly Coded Applications
  - What else?

Benefits

- Shields from …
  - Casual Attackers
  - Automated Assaults
  - Oblique Threats
- Protects …
  - Unmanaged
  - Unpatched
  - Unhardened
- Defeats … canned exploits

What's Next

- More Platforms
  - Open-Source Router Firmware
  - Linux-Based Switches
- Production Trials
- Talk to vendors

Final Thoughts

- Accurate target identification is key to a successful attack
- Identification that is way too easy for an attacker to perform
- Let’s change that with fingerprint prevention
- I’ve proven that it can be done
- Now, we just have to make it happen

Proof of Concept

IDGuard v0.50 for Linux-Based Networking

- Network-Wide Fingerprint Prevention
- IPv4, and TCP normalizations
- Authorized Activity Exclusions
- Linux Kernel Module Implemenation

IDGuard v0.60 for Linux-Based Networking

- Adds IPv6 Support
- Coming Next Month!

SHA1 hash is **289256c1b46f7f7443527364ad4a75ee0a072160**
Updates can be found at http://idguard.sourceforge.net/

Links

- http://www.wisegeek.com/what-is-packet-mangling.htm
- http://www.openbsd.gr/faq/pf/scrub.html
- http://www.linuxsecurity.com.br/info/fw/PacketManglingwithiptables.doc
- http://chdir.org/~nico/scrub/
- http://www.cisco.com/en/US/docs/security/asa/asa82/configuration/guide/conns_tcpnorm.pdf
- http://www.cisco.com/en/US/docs/interfaces_modules/services_modules/ace/v3.00_A2/configuration/security/guide/tcpipnrm.pdf
- http://www.sans.org/reading_room/whitepapers/intrusion/packet-level-normalisation_1128
- http://nmap.org/book/osdetect-methods.html
- http://rcp100.sourceforge.net
- http://wiki.openwrt.org/toh/mikrotik/rb450g
- http://wiki.openwrt.org/doc/howto/buildroot.exigence
- http://wiki.openwrt.org/doc/howto/build
- http://wiki.openwrt.org/doc/howto/generic.flashing
- http://wiki.openwrt.org/doc/devel/crosscompile

Special Thanks

- Aditiya Sood
- Kenny Nguyen and E-CQURITY
- Kevin Fogarty
- Kathy Gillette
- Nick Pruitt

Q&A