ARSENIC MOBILE

A configuration free ARP poisoning and credential extraction tool for iOS

SSL/TLS – A BRIEF HISTORY

“You have to know the past to understand the present.” – Dr. Carl Sagan

SSL/TLS BASICS

SSL

- "Secure Sockets Layer"
- Uses asymetric cryptography to establish symetric session keys to encrypt traffic
- Research on transport layer security began in 1993[1]
- Originally developed in 1995 by Netscape[2]
- Completely rewritten in 1996 (v 3.0)[3]
- Replaced by TLS

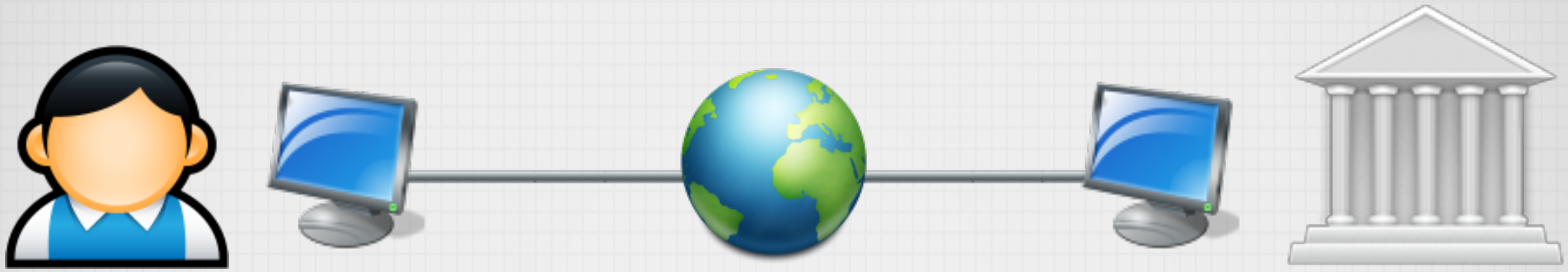
TLS

- "Transport Layer Security"
- Uses the same basic cryptography as SSL
- 1.0 released in 1999 to succeed SSL
- All major browsers support 1.0, but only a few support 1.1 or 1.2
- Most websites are still using 1.0, despite a published attack (BEAST)[4,5]

SSL/TLS USAGE — AN EXAMPLE

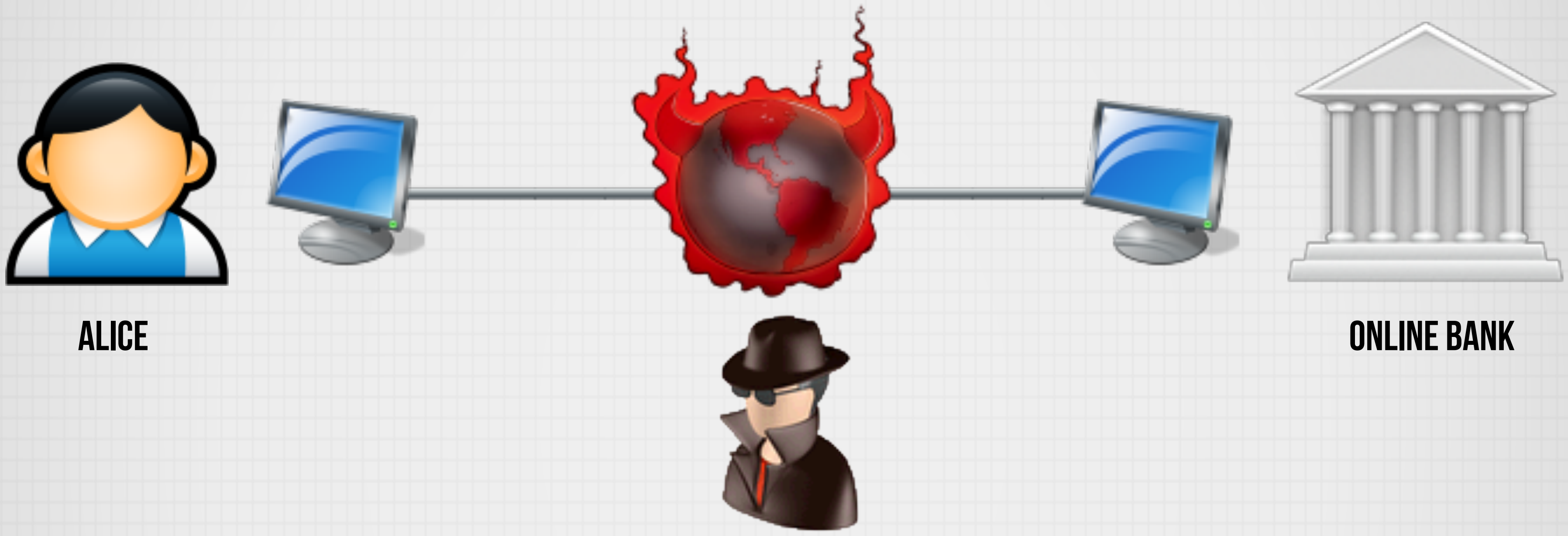
“Make things as simple as possible, but not simpler.” – Albert Einstein

PRACTICAL EXAMPLE



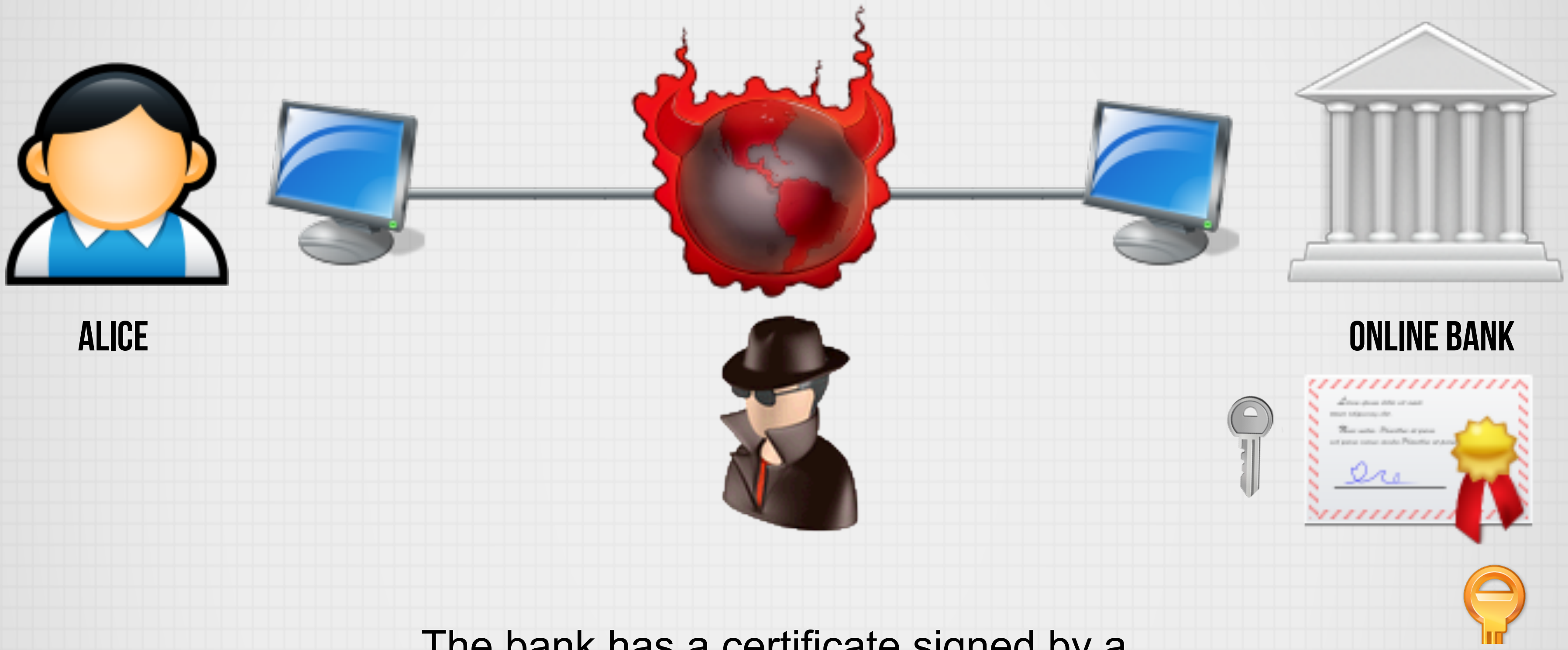
Alice wants to do some secure online banking…

PRACTICAL EXAMPLE



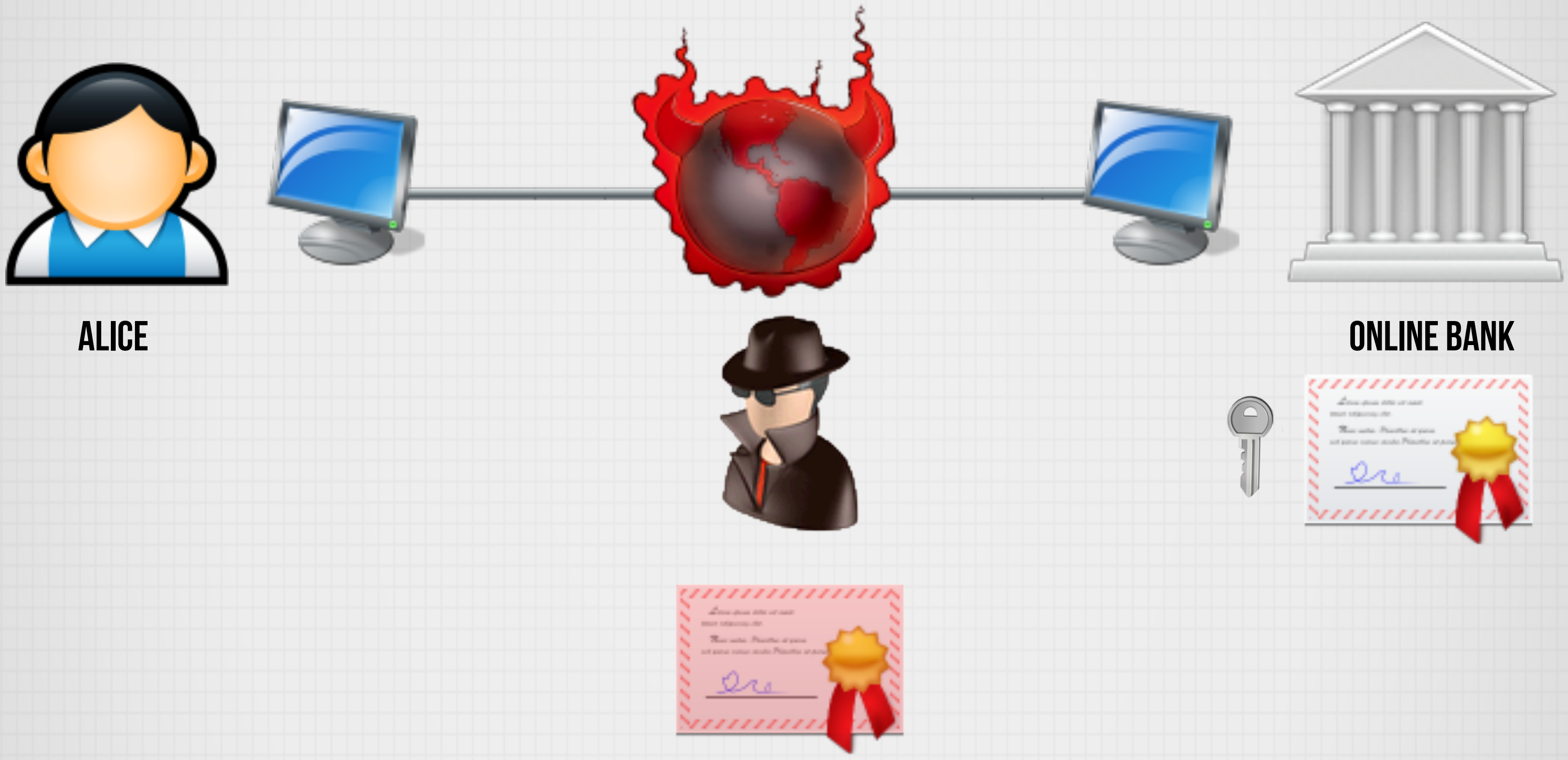
But the internet is a nasty place.

PRACTICAL EXAMPLE



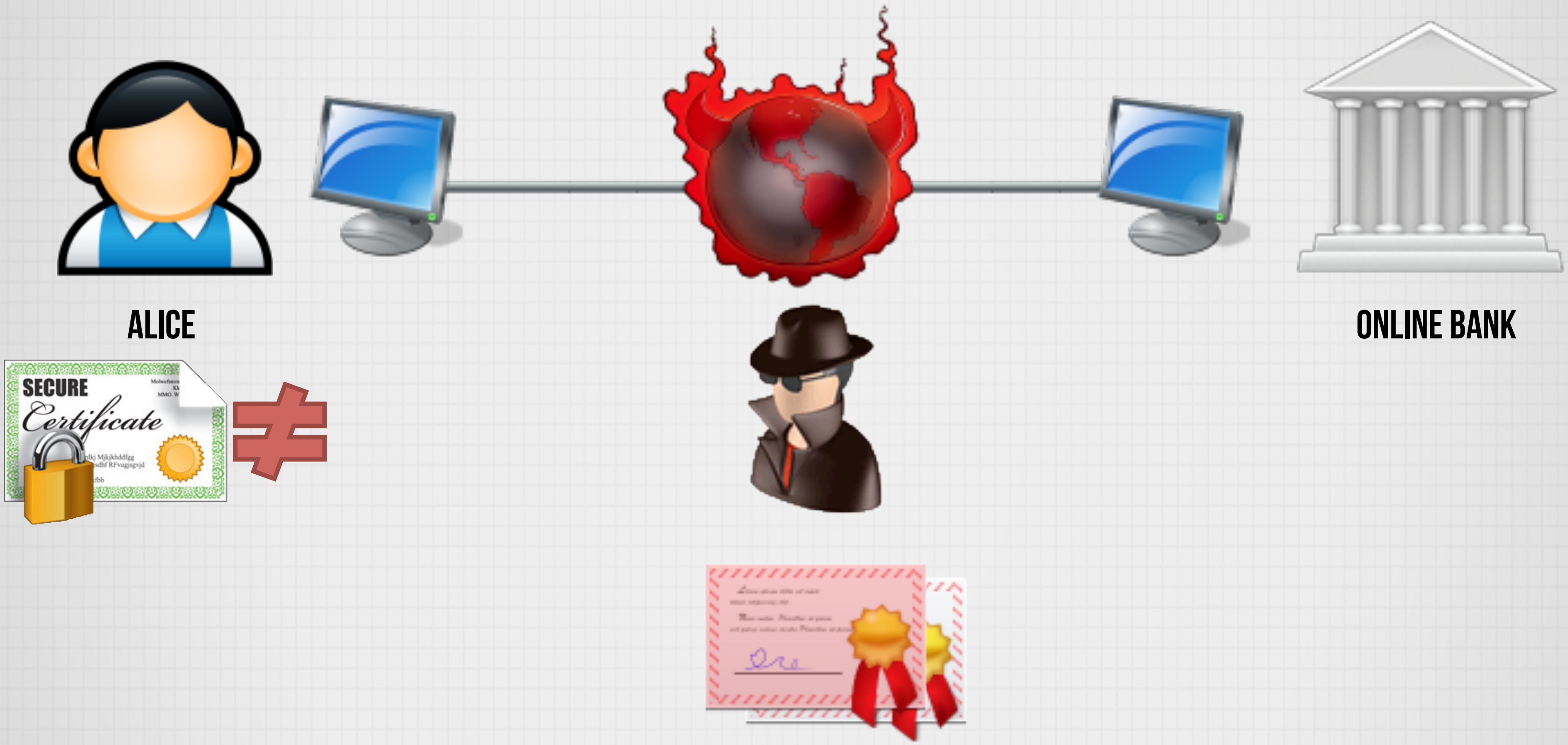
The bank has a certificate signed by a certificate authority (i.e. VeriSign). The bank generates its half of the session key and sends it with the certificate.

PRACTICAL EXAMPLE



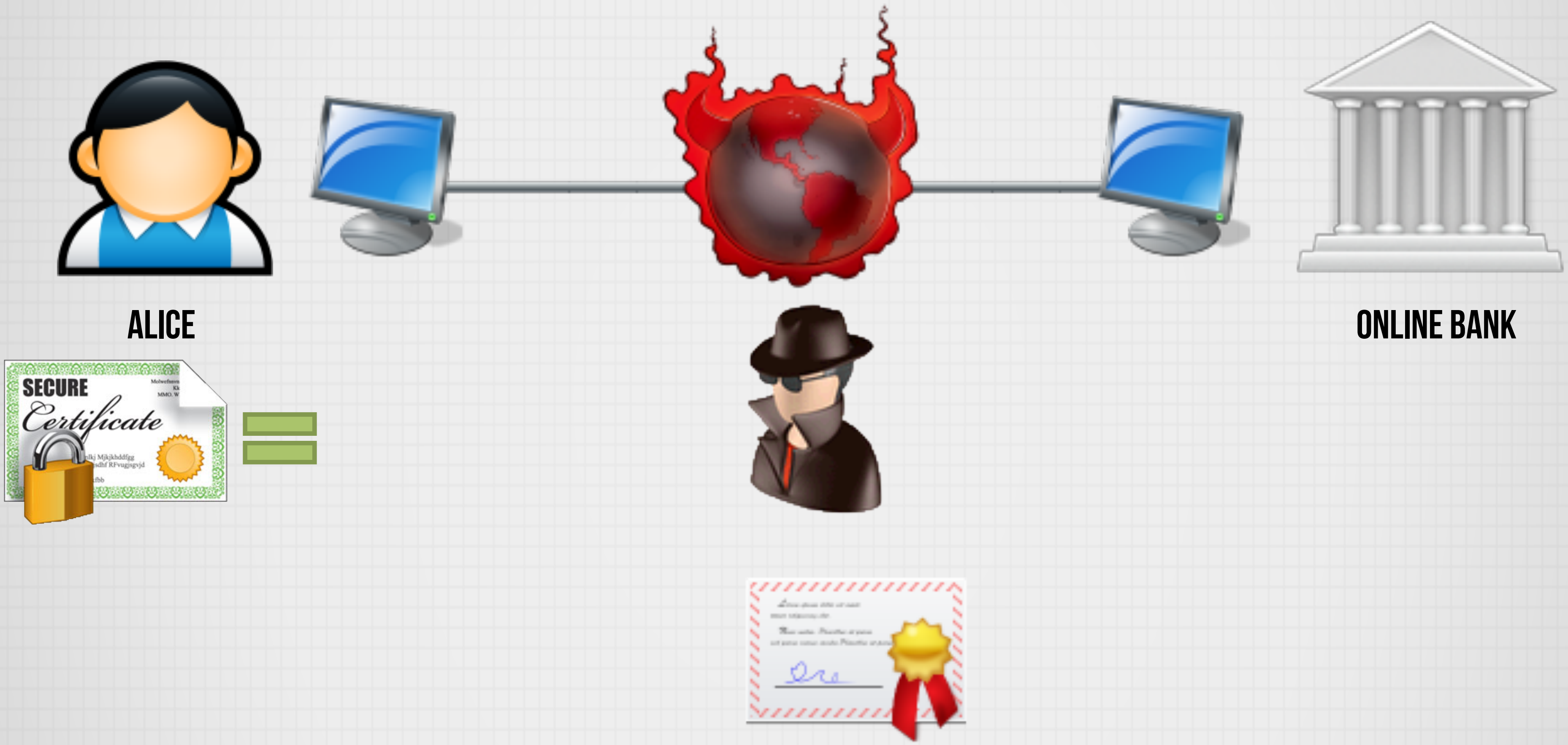
A man-in-the-middle attacker could replace this certificate with his own…

PRACTICAL EXAMPLE



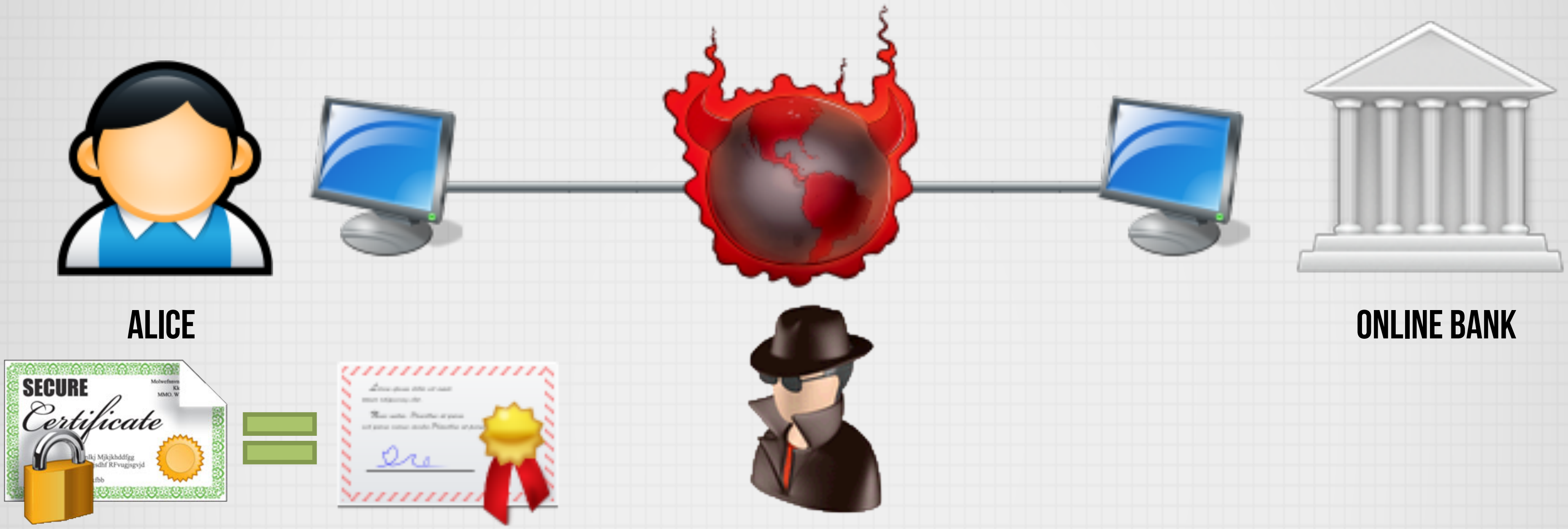
But this certificate won't pass the check against the certificate authorities root certificate.

PRACTICAL EXAMPLE



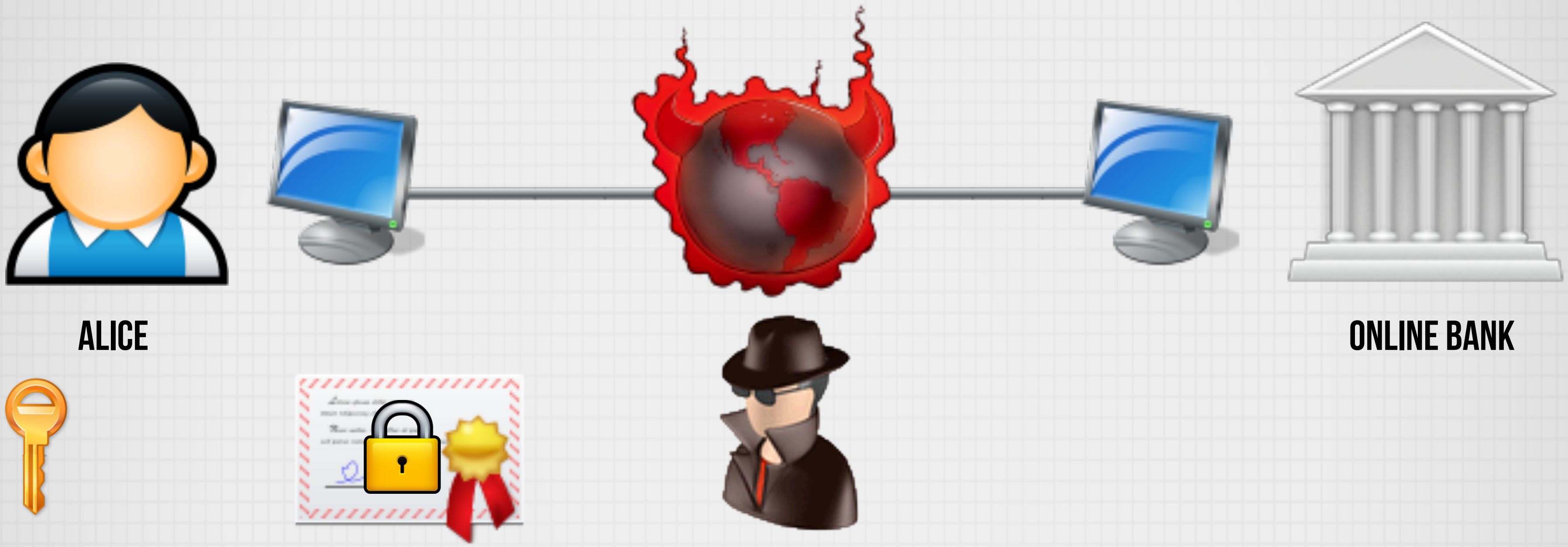
In order for Alice to continue the handshake, the correct certificate must be passed along.

PRACTICAL EXAMPLE



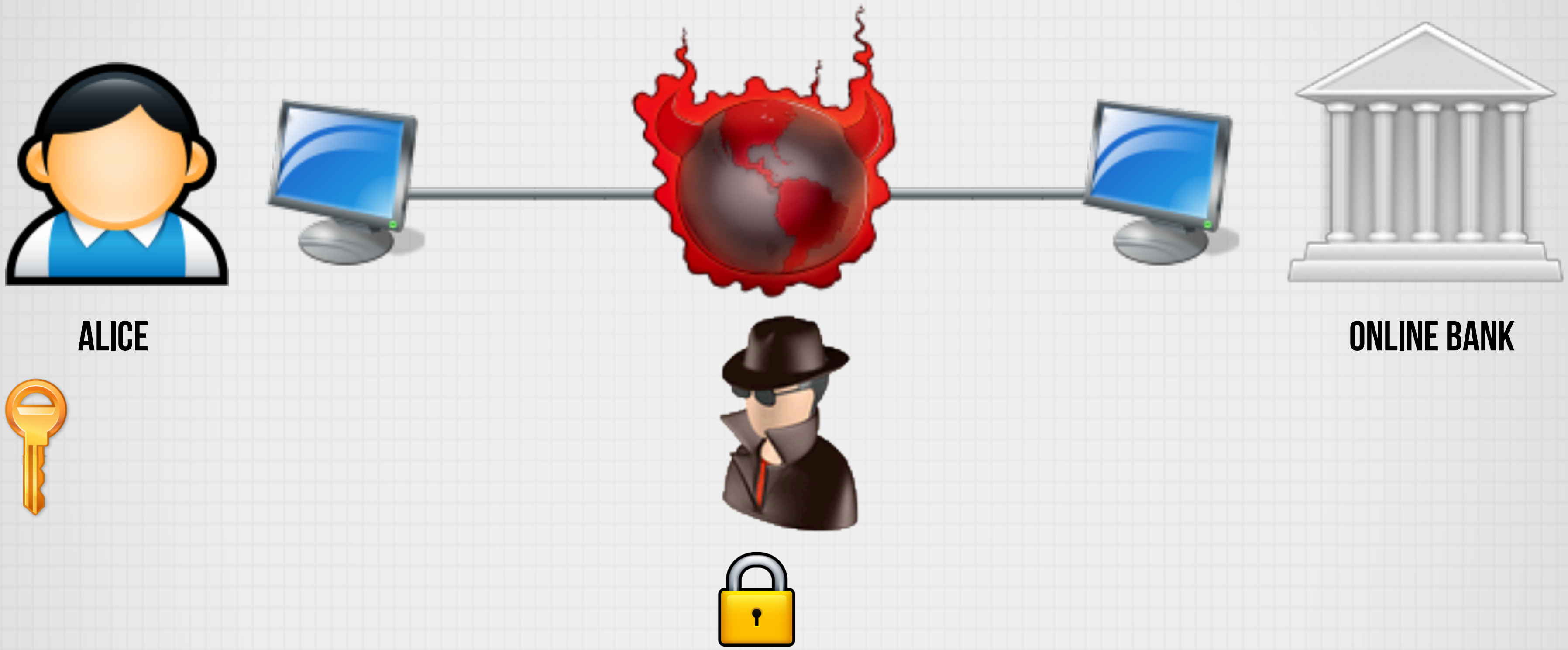
Alice verifies the bank's certificate with her copy of the certificate authorities root certificate. She then extracts part of the symmetric key that was sent with the certificate that her session will use.

PRACTICAL EXAMPLE



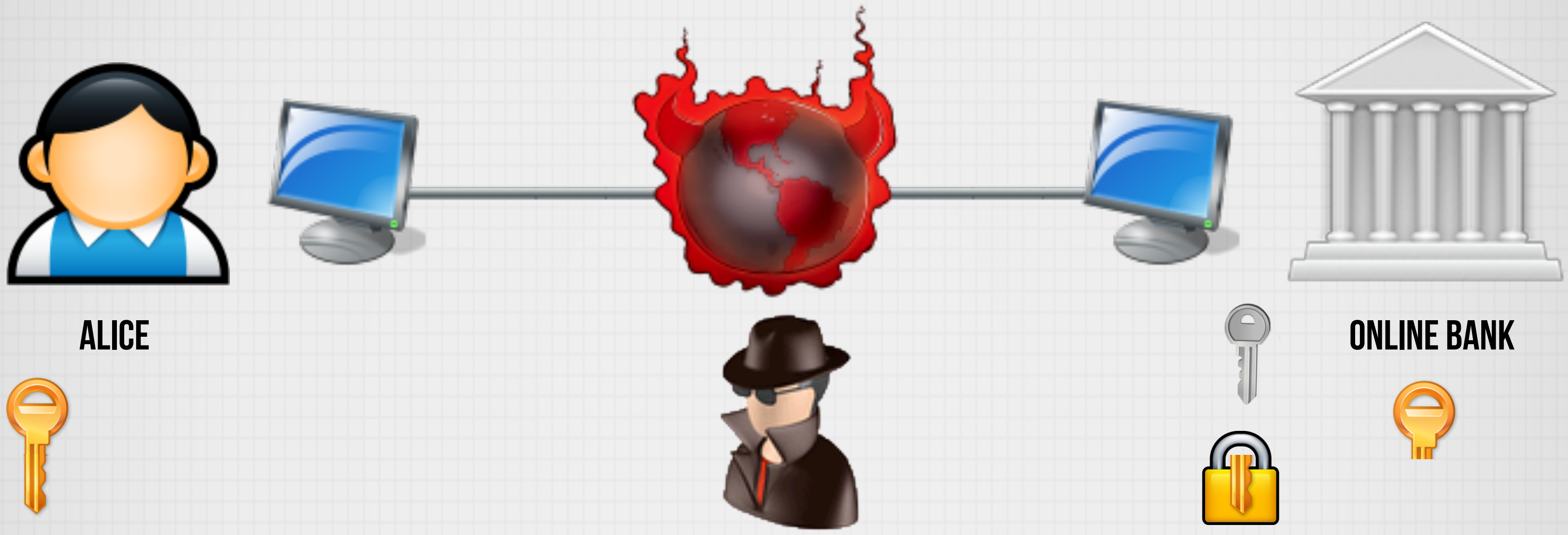
Alice generates the other half of the key and sends it to her bank after encrypting it with the bank's public key (from the certificate). In reality this is a Diffie-Hellman key exchange, usually RSA.

PRACTICAL EXAMPLE



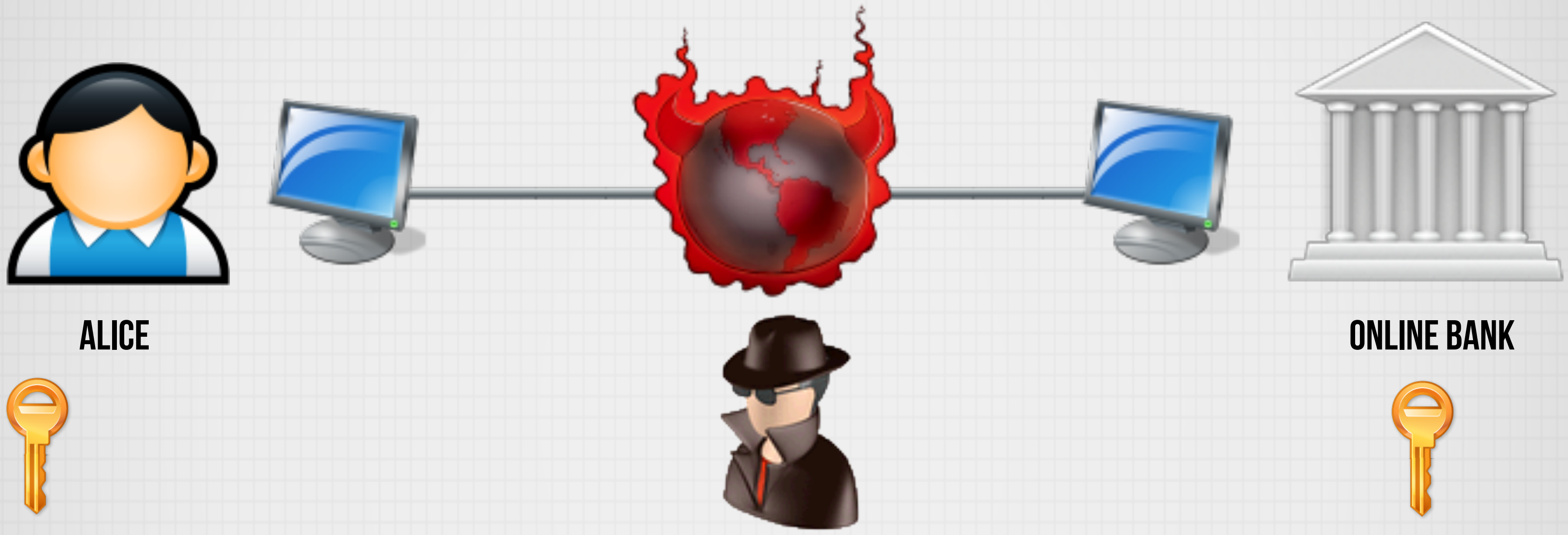
The man-in-the-middle attacker can’t decrypt the key because he doesn’t have the bank’s private key.

PRACTICAL EXAMPLE



The bank now decrypts the second part of the key and creates the session key.

PRACTICAL EXAMPLE



Now both Alice and the bank have a session key that they can use to encrypt messages with a symmetric cypher and any man-in-the-middle attacker will be unable to read them.

SO... HOW CAN WE BREAK IT?

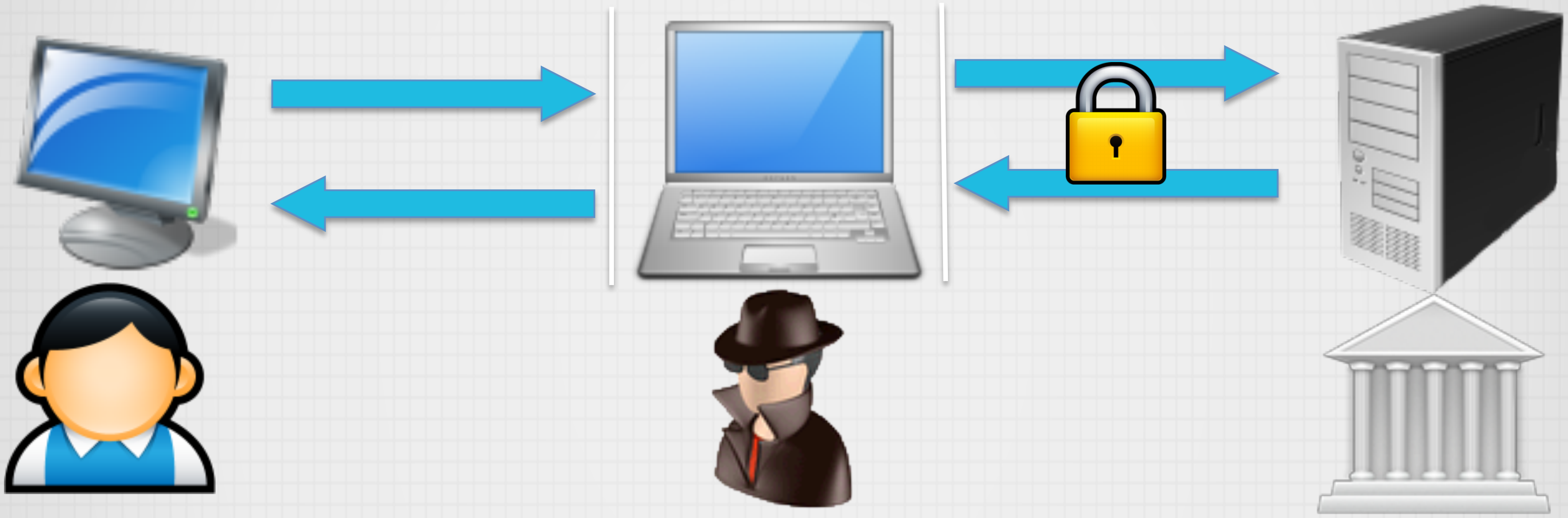
“Any sufficiently advanced technology is indistinguishable from magic.” – Arthur C. Clarke

POSSIBLE ATTACK VECTORS

- Determine the private key associated with a certificate (really hard)
- Sign your own certificate to say you are someone else (also really hard)
- Break into a certificate authority and sign yourself a bunch of certificates (really hard, but it has been done)[6]
- But what if we never let the user get to SSL/TLS?
  - Users most often encounter SSL/TLS as a result of:
    - Hyperlinks
    - HTTP 302 redirects
  - In other words, users get to HTTPS via HTTP
    - How secure is HTTP?
    - Links and 302's are susceptible to a regular HTTP man-in-the-middle attack

SSLSTRIP

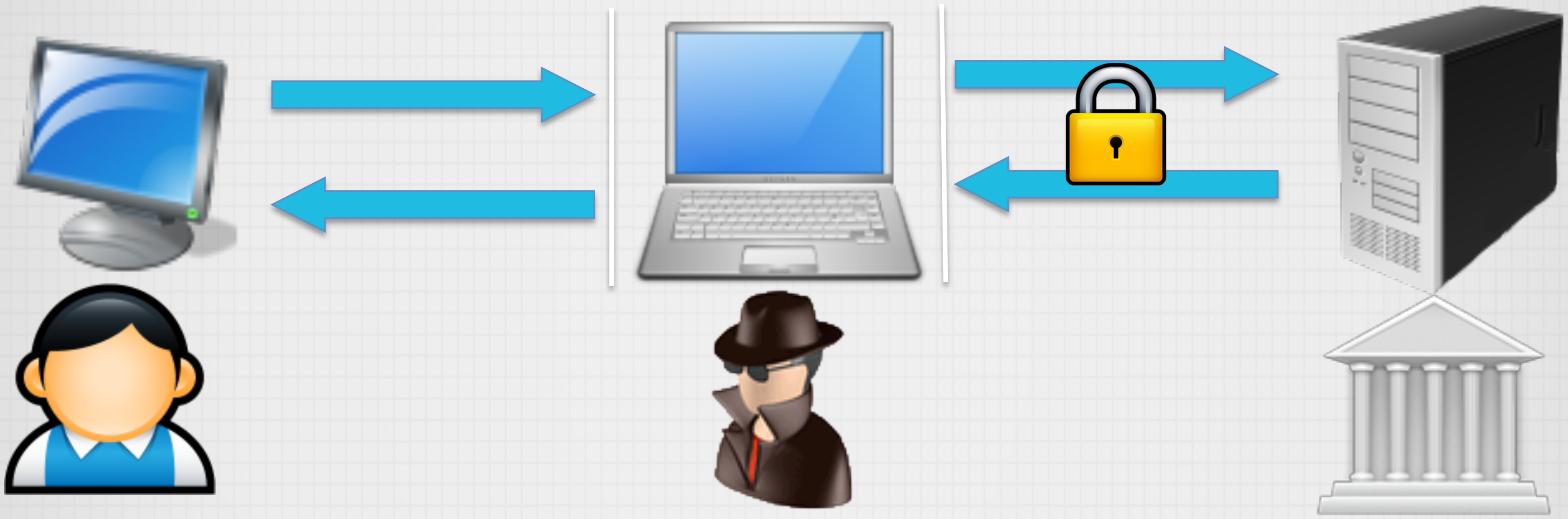
- Released in 2009 by Moxie Marlinspike, this tool does just what we want.[7]



- When the victim makes an HTTP request, sslstrip connects out to the server on HTTPS, and dynamically replaces all instances of “https” with “http” in the return traffic.
- The server can’t tell the difference, the user is never presented with “This connection is untrusted” warnings, and sslstrip can see all the traffic.

ARSENIC MOBILE

- Arsenic takes the hassle out of setting up an sslstrip attack.[8]
- "Zero configuration"



- Having an easy to use attack on a laptop is cool…
- But what about an iPhone?

GOING MOBILE - CHALLENGES

“Never trust a computer you can’t throw out a window.” – Steve Wozniak

ARSENIC MOBILE – WHAT WE NEED

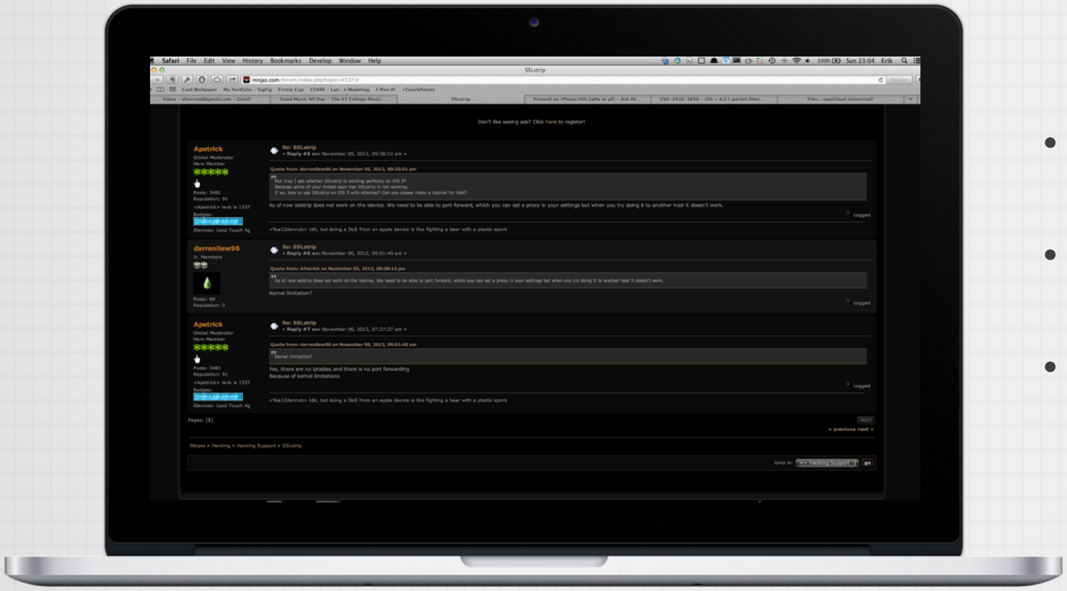
LINUX/OSX

- nmap
- Python+Scapy or arpspoof
  - To setup the MITM attack
- Iptables or ipfw
  - To redirect our traffic to sslstrip
- sslstrip
  - Python
  - Twisted-Web
    - zope-interface
  - pyOpenSSL
- tail to watch the sslstrip log
- A parser to pull out credentials

IOS

nmap	✓
Python	✓
Scapy	✗
arpspoof	✓
iptables	✗
ipfw	✗
sslstrip	✗
Twisted-Web	✗
zope-interface	✗
pyOpenSSL	✗
tail	✗
Parser (python)	✓

HAS ANYONE TRIED THIS BEFORE?



Don't like seeing ads? Click here to register!

Apetrick
Global Moderator
Hero Member
Posts: 3481
Reputation: 91
<Apetrick> lank is 1337
Badges: Global Moderator
iDevices: Ipod Touch 4g

Re: SSLstrip
« Reply #5 on: November 05, 2012, 09:38:12 pm »

Quote from: darrenliew96 on November 05, 2012, 09:25:51 pm
> But may I ask whether SSLstrip is working perfectly on iOS 5?
> Because some of your thread says that SSLstrip in not working.
> If so, how to use SSLstrip on iOS 5 with ettercap? Can you please make a tutorial for that?

As of now sslstrip does not work on the idevice. We need to be able to port forward, which you can set a proxy in your settings but when you try doing it to another host it doesn't work.

Logged

<%a12danrulz> Idk, but doing a DoS from an apple device is like fighting a bear with a plastic spork

darrenliew96
Jr. Members
Posts: 89
Reputation: 2

Re: SSLstrip
« Reply #6 on: November 06, 2012, 05:51:40 am »

Quote from: APetrick on November 05, 2012, 09:38:12 pm
> As of now sslstrip does not work on the idevice. We need to be able to port forward, which you can set a proxy in your settings but when you try doing it to another host it doesn't work.

Kernel limitation?

Logged

Apetrick
Global Moderator
Hero Member
Posts: 3481
Reputation: 91
<Apetrick> lank is 1337
Badges: Global Moderator
iDevices: Ipod Touch 4g

Re: SSLstrip
« Reply #7 on: November 06, 2012, 07:27:27 am »

Quote from: darrenliew96 on November 06, 2012, 05:51:40 am
> Kernel limitation?

Yes, there are no iptables and there is no port forwarding
Because of kernel limitations

Logged

<%a12danrulz> Idk, but doing a DoS from an apple device is like fighting a bear with a plastic spork

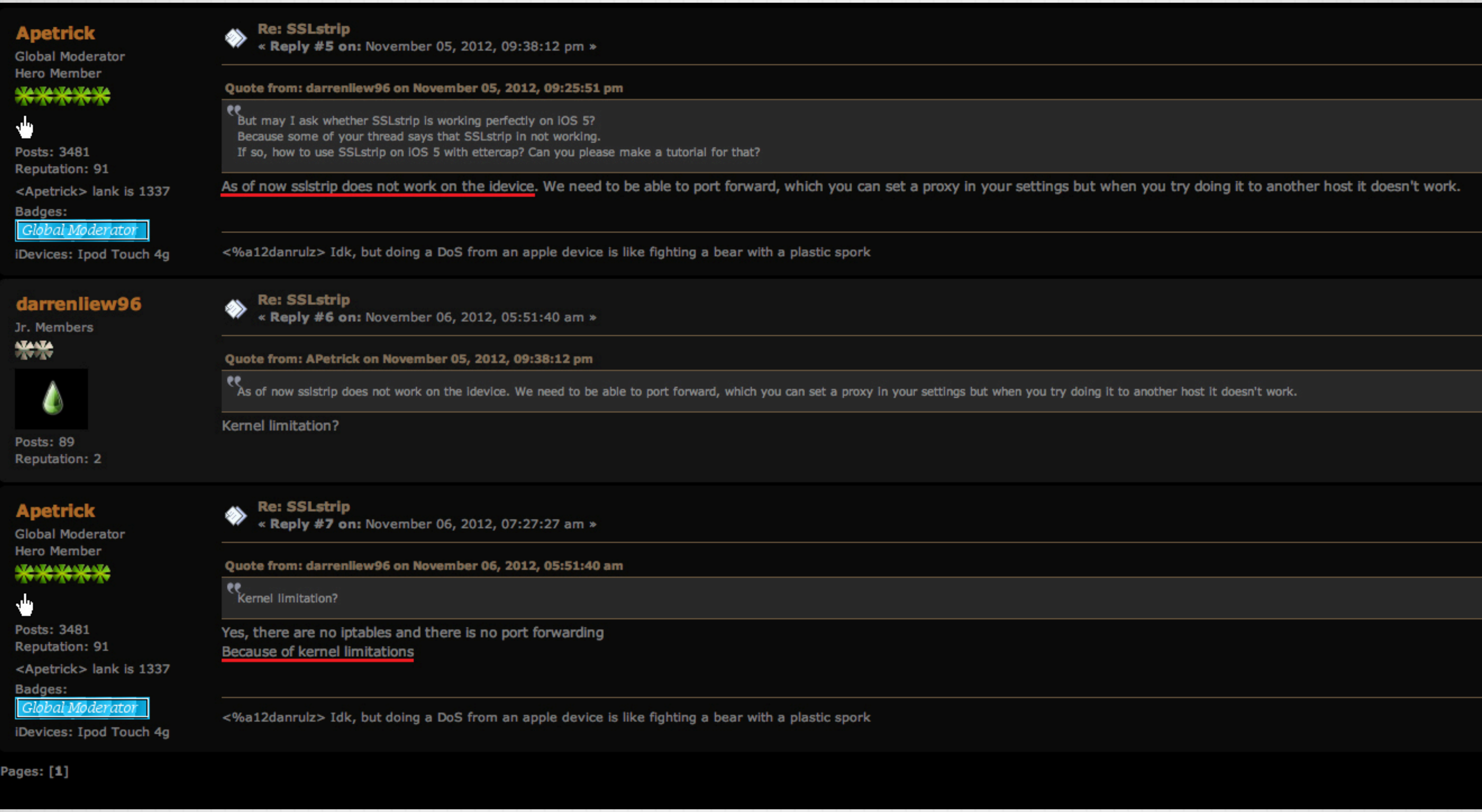
Pages: [1]

« previous next »

iNinjas » Hacking » Hacking Support » SSLstrip

Jump to: => Hacking Support go

- Lots of work has been done in jailbroken iOS development
- Python, Ruby, even Metasploit has been ported to iOS
- Why not sslstrip?


Apetrick
Global Moderator
Hero Member
Posts: 3481
Reputation: 91
<Apetrick> lank is 1337
Badges: Global Moderator
iDevices: Ipod Touch 4g

Re: SSLstrip
« **Reply #5 on:** November 05, 2012, 09:38:12 pm »

Quote from: darrenliew96 on November 05, 2012, 09:25:51 pm

> But may I ask whether SSLstrip is working perfectly on iOS 5?
> Because some of your thread says that SSLstrip in not working.
> If so, how to use SSLstrip on iOS 5 with ettercap? Can you please make a tutorial for that?

As of now sslstrip does not work on the idevice. We need to be able to port forward, which you can set a proxy in your settings but when you try doing it to another host it doesn't work.

<%a12danrulz> Idk, but doing a DoS from an apple device is like fighting a bear with a plastic spork

darrenliew96
Jr. Members
Posts: 89
Reputation: 2

Re: SSLstrip
« **Reply #6 on:** November 06, 2012, 05:51:40 am »

Quote from: APetrick on November 05, 2012, 09:38:12 pm

> As of now sslstrip does not work on the idevice. We need to be able to port forward, which you can set a proxy in your settings but when you try doing it to another host it doesn't work.

Kernel limitation?

Apetrick
Global Moderator
Hero Member
Posts: 3481
Reputation: 91
<Apetrick> lank is 1337
Badges: Global Moderator
iDevices: Ipod Touch 4g

Re: SSLstrip
« **Reply #7 on:** November 06, 2012, 07:27:27 am »

Quote from: darrenliew96 on November 06, 2012, 05:51:40 am

> Kernel limitation?

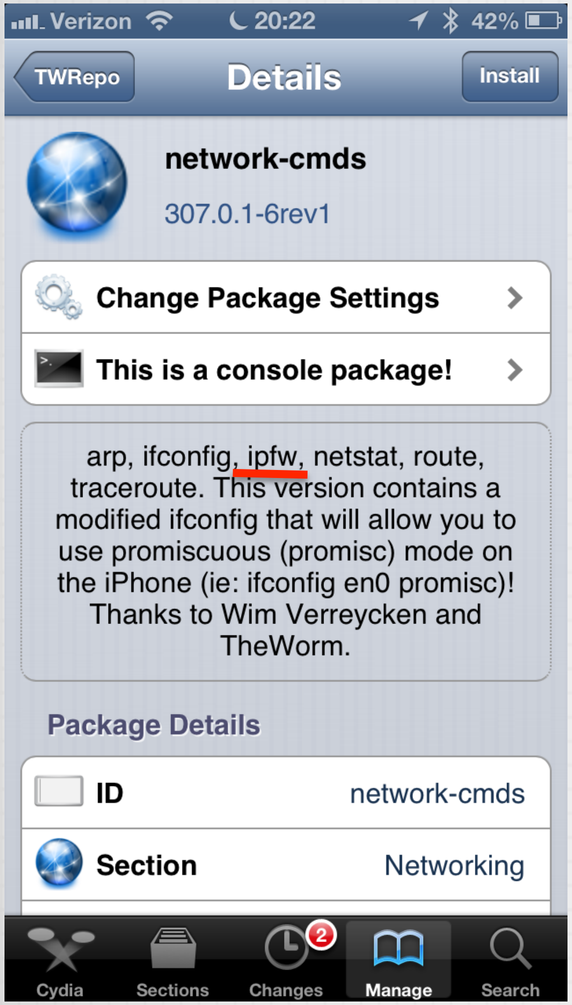
Yes, there are no iptables and there is no port forwarding
Because of kernel limitations

<%a12danrulz> Idk, but doing a DoS from an apple device is like fighting a bear with a plastic spork

Pages: [1]

ARSENIC MOBILE

BUT WAIT! IPFW!



network-cmds

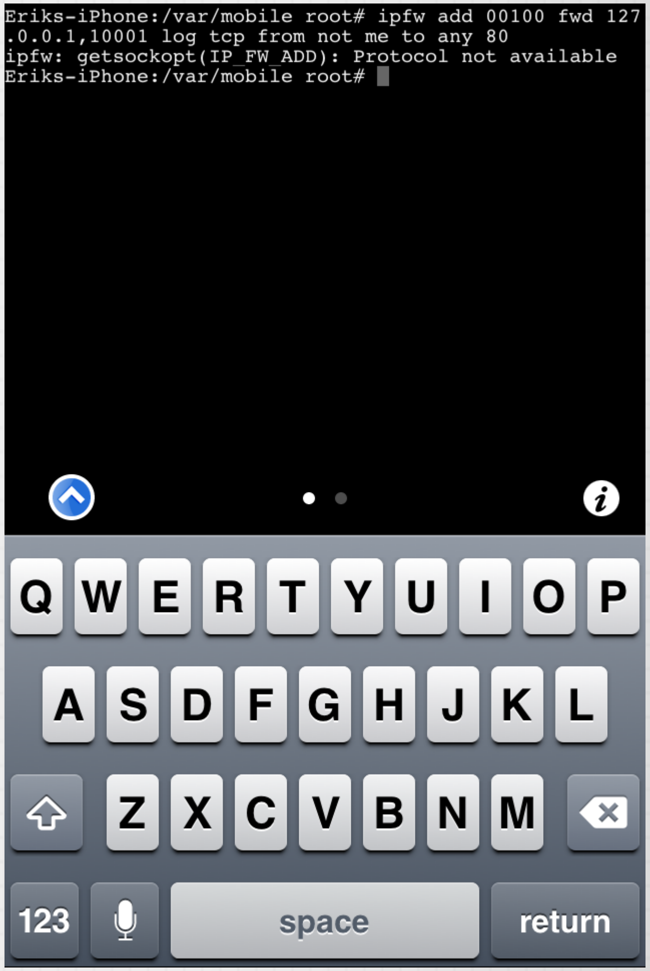
307.0.1-6rev1

arp, ifconfig, ipfw, netstat, route, traceroute. This version contains a modified ifconfig that will allow you to use promiscuous (promisc) mode on the iPhone (ie: ifconfig en0 promisc)! Thanks to Wim Verreycken and TheWorm.

Package Details

ID	network-cmds
Section	Networking

If it sounds to good to be true….



```
Eriks-iPhone:/var/mobile root# ipfw add 00100 fwd 127
.0.0.1,10001 log tcp from not me to any 80
ipfw: getsockopt(IP_FW_ADD): Protocol not available
Eriks-iPhone:/var/mobile root# 
```

It probably is.

ARSENIC MOBILE

ALL HOPE IS NOT LOST

- The iOS 4.0 jailbreak was based on a vulnerability in the implementation of the BSD pf packet filter.[10]
- Why has no one else tried to use the pf packet filter for sslstrip?
  - Beats me
- Challenges remain
  - Configuring pf
  - Porting sslstrip and dependencies
    - Requires compiling C source code on iOS
    - Some things just don’t work (but aren’t critical for this application)
  - Write a tail function in python

WHAT WAS INSTALLED

- MobileTerminal and OpenSSH from Cydia (standard packages)
- GCC and dependencies (compiled for iOS)
- Python 2.7.3 (compiled for iOS)
  - Custom Makefile
- zope-interface 4.05 (from source, pure python)
- Twisted-web 13.0.0 (from source, python and C)
  - portmap extension excluded
- pyOpenSSL 0.11 (from source, python and C)
- sslstrip 0.9
  - Modified to eliminate two error outputs
- dsniff 2.4b1 (compiled for iOS)
- ~~nmap (compiled for iOS)~~ Integrated scanner built in python
- pf.os (taken from OSX)
- Custom pf.conf
- arsenic-mobile

LIVE DEMO

“Computers are like Old Testament gods; lots of rules and no mercy.” – Joseph Campbell

FUTURE WORK

- Package everything into a .deb for easy distribution
- Improve the parser
- Improve killsessions to work with sites that keep the entire session in SSL
- GUI

QUESTIONS?

"Defect-free software does not exist." – Wietse Venema

WORKS CITED

1. Thomas Y. C. Woo, Raghuram Bindignavle, Shaowen Su and Simon S. Lam, *SNP: An interface for secure network programming* Proceedings USENIX Summer Technical Conference, June 1994
2. "THE SSL PROTOCOL". Netscape Corporation. 2007. Archived from the original on 14 June 1997.
3. Freier, A., Karlton, P., Kocher, P.: The Secure Sockets Layer (SSL) Protocol Version 3.0. RFC 6101 (Historic) (Aug 2011), http://www.ietf.org/rfc/rfc6101.txt
4. "SSL Pulse: Survey of the SSL Implementation of the Most Popular Web Sites". Retrieved 2013-04-26.
5. Dan Goodin (2011-09-19). "Hackers break SSL encryption used by millions of sites".
6. "Report of incident on 15-MAR-2011". Comodo group. Retrieved 2011-03-24.
7. Marlinspike, Moxie, "Defeating SSL In Practice" Blackhat 2009 http://www.blackhat.com/presentations/bh-dc-09/Marlinspike/BlackHat-DC-09-Marlinspike-Defeating-SSL.pdf
8. https://github.com/kernel-sanders/Arsenic
9. http://ininjas.com/forum/index.php?topic=4537.0
10. http://esec-lab.sogeti.com/post/2010/12/09/CVE-2010-3830-iOS-4.2.1-packet-filter-local-kernel-vulnerability

All icons from www.iconarchive.com