

RSA[®]Conference2015

San Francisco | April 20-24 | Moscone Center

SESSION ID: STR-T08

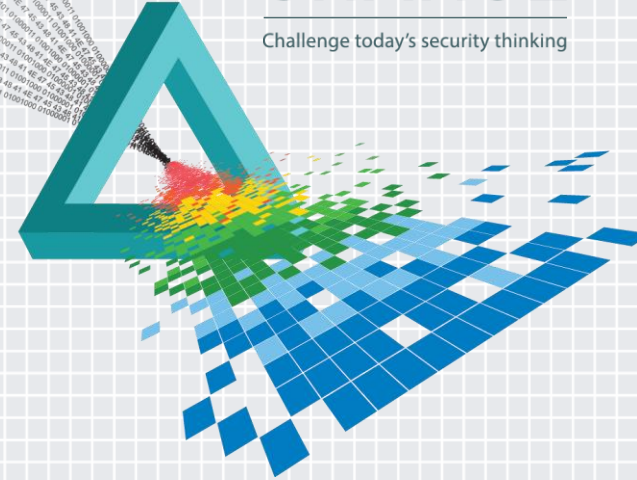
Software Supply Chains: Another Bug Bites the Dust.

Todd Inskeep

Global Security Assessments VP
Samsung Business Services
@Todd_Inskeep

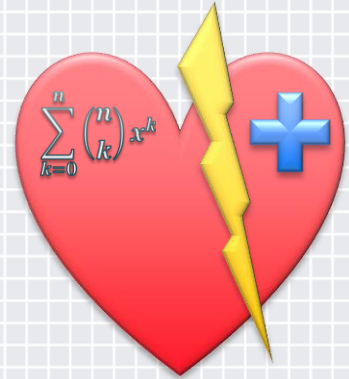
CHANGE

Challenge today's security thinking

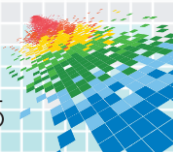


Series of Recent, Large, Long-term Security Issues

- ◆ Heartbleed – April 2014 (1998)
- ◆ LZO – June 2014 (1994)
- ◆ Shellshock- September 2014 (1989)
- ◆ POODLE – October 2014 (1996) [December 2014 variant for TLS]
- ◆ Winshock - November 2014 (1996)
- ◆ Kerberos Checksum Vulnerability – November 2014 (~2000)
- ◆ Equation Group – February 2015 (~2001)
- ◆ FREAK – March 2015 (~1998)



How Vulnerable is the Internet?



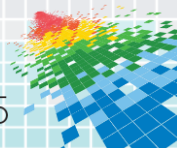
How Vulnerable is the Internet?

Bigger than the biggest thing ever and then some. Much bigger than that in fact, really amazingly immense, a totally stunning size, real "wow, that's big," time. ...Gigantic multiplied by colossal multiplied by staggeringly huge is the sort of concept we're trying to get across here.



- Hitchhiker's Guide to the Galaxy

How Vulnerable is your company?



Look at ONE company - Yours



OPEN SOURCE
AUDIT STATISTICS



90% of code bases contain
undisclosed open source



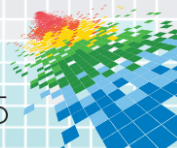
75% of audits contain
unknown licenses

50% of code audits contain
GPL



<https://www.blackducksoftware.com/>

- ◆ **Establish** Ownership for Software Security
- ◆ **Identify** Critical Software – Build a BOM
 - ◆ **Evaluate** FOSS & Licenses/Management
- ◆ **Scan** Systems and Codebase
 - ◆ Compare Codebases to NVD
- ◆ **Establish** SDLC for Software Security
 - ◆ **Consider** the Building Security In Maturity Model (BSIMM)
- ◆ **Review** Code Repositories & Governance
- ◆ **Conduct** Supplier Software Assessment
- ◆ **Implement** Checks on Firmware
- ◆ **Monitor** the changing Environment



Establish Ownership for Software Security

- ◆ **Responsible**
- ◆ **Accountable**
- ◆ **Resourced**
- ◆ **Empowered with Authority**

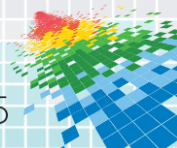


"If everyone is in charge, no one's in charge."

Aligning the Stars: How to Succeed when Professionals Drive Results
By Jay William Lorsch, Thomas J. Tierney

Everyone is responsible and no one is to blame.

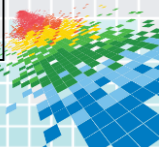
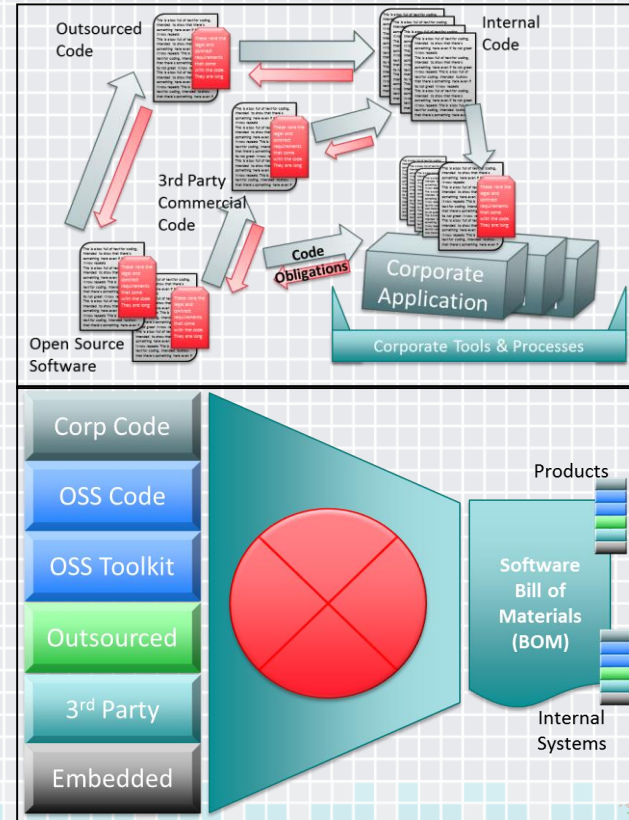
- Will Schultz, American Economist



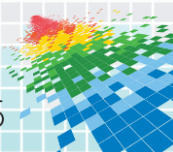
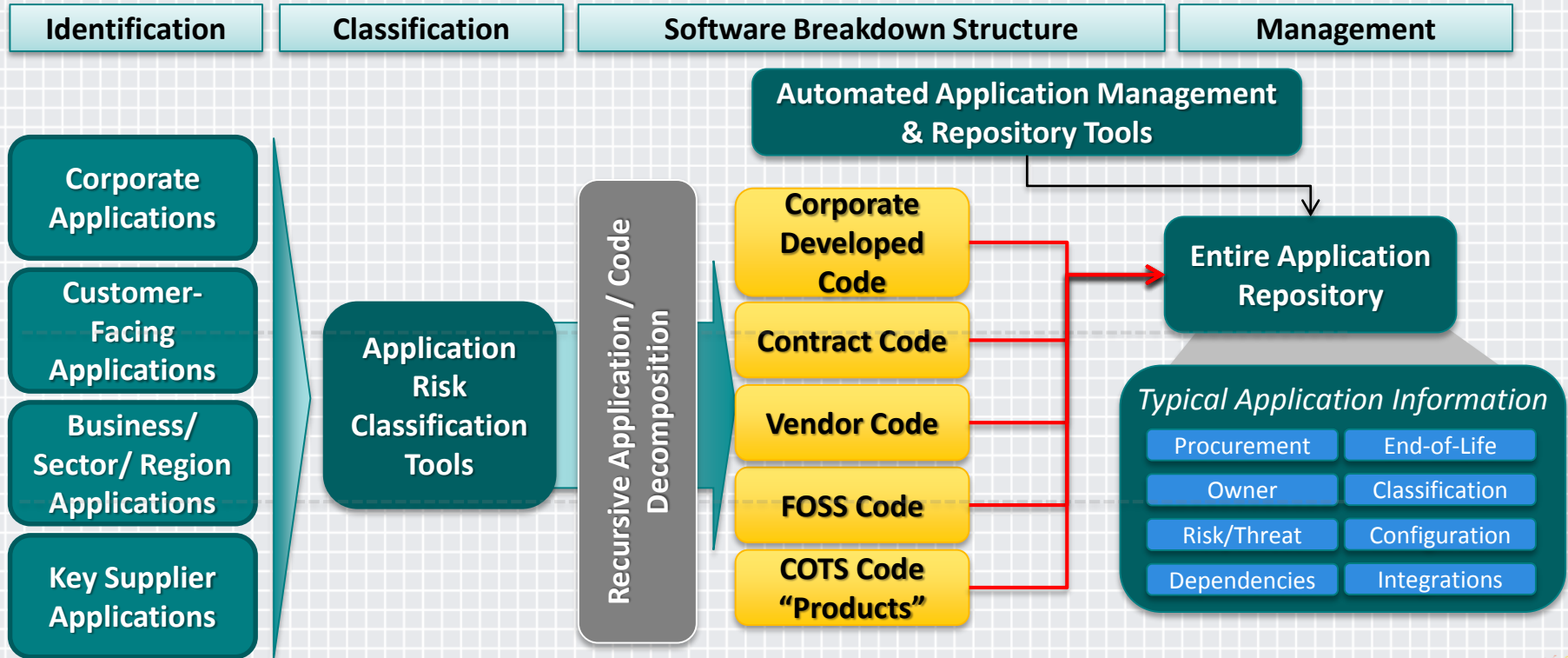
Identify Critical Software – Document the BOM

- ◆ Source(s) of Code
 - ◆ FOSS Code
 - ◆ Licensed Code
 - ◆ Developed Code
 - ◆ Firmware

- ◆ Code in Tools
- ◆ Code in Frameworks and APIs
- Evaluate FOSS Licenses & Management
 - ◆ Software Package Data Exchange ®

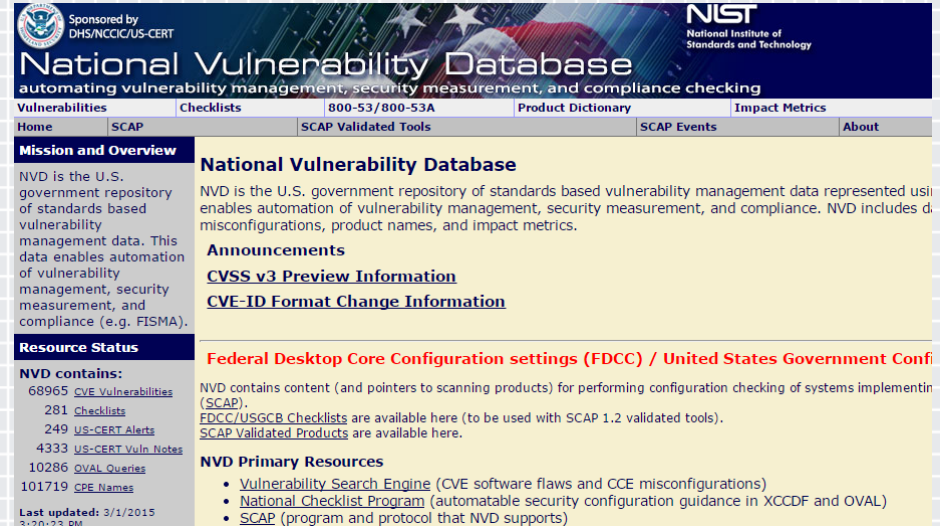


Building the Software BOM



Scan Systems and Codebases

- ◆ Audit Tools
- ◆ Static Analysis Tools
- ◆ Dynamic Analysis Tools
- ◆ Red Team / Active Penetration Testing
- ◆ Compare Codebases to NVD
 - ◆ National Vulnerability Database - <https://nvd.nist.gov/>
 - ◆ Authoritative Source for most vulnerabilities and threat tracking
 - ◆ Standardizes naming conventions & references for searchability
 - ◆ Assesses threat level of vulnerabilities
 - ◆ Limited information on status of vulnerability repair



Sponsored by
DHS/NCCIC/US-CERT

NIST
National Institute of
Standards and Technology

National Vulnerability Database

automating vulnerability management, security measurement, and compliance checking

Vulnerabilities Checklists 800-53/800-53A Product Dictionary Impact Metrics

Home SCAP SCAP Validated Tools SCAP Events About

Mission and Overview

NVD is the U.S. government repository of standards based vulnerability management data. This data enables automation of vulnerability management, security measurement, and compliance (e.g. FISMA).

Resource Status

NVD contains:

- 68965 [CVE Vulnerabilities](#)
- 281 [Checklists](#)
- 249 [US-CERT Alerts](#)
- 4333 [US-CERT Vuln Notes](#)
- 10286 [OVAL Queries](#)
- 101719 [CPE Names](#)

Last updated: 3/1/2015

National Vulnerability Database

NVD is the U.S. government repository of standards based vulnerability management data represented using automation of vulnerability management, security measurement, and compliance. NVD includes data on misconfigurations, product names, and impact metrics.

Announcements

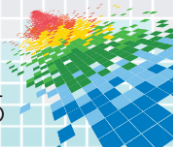
- [CVSS v3 Preview Information](#)
- [CVE-ID Format Change Information](#)

Federal Desktop Core Configuration settings (FDCC) / United States Government Conf

NVD contains content (and pointers to scanning products) for performing configuration checking of systems implementing (SCAP).
FDCC/USGCB Checklists are available here (to be used with SCAP 1.2 validated tools).
SCAP Validated Products are available here.

NVD Primary Resources

- [Vulnerability Search Engine](#) (CVE software flaws and CCE misconfigurations)
- [National Checklist Program](#) (automatable security configuration guidance in XCCDF and OVAL)
- [SCAP](#) (program and protocol that NVD supports)



Coverity Scan – Static Analysis of FOSS

3900+ Projects on Coverity Scan

Did you know reaction of Apache Tomcat committer when he looked at the defects found by Coverity?
"Wow, this is great. I've used FindBugs before both inside and outside of ASF projects, but this is ... just amazing."

Interested in a specific programming language?
 • Java • C/C++ • C#

Project	Lines of code analyzed	Language
Broadcom Linux Kernel Open Source Repository	pending build	C/C++
Jlighthouse/util-linux	146,109	C
LeesLinuxKernel	34,530	C
Linux	9,478,601	C
LinuxCNC	353,700	C
Linux-HA	122,886	C
LinuxMatt/IronGrip	pending	C

Linux

Project Name: Linux
 Lines of code analyzed: 9,478,601
 On Coverity Scan since: Feb 24, 2006
 Last build analyzed: 6 days ago

Language: C/C++
 Repository URL: <http://git.kernel.org/>
 Homepage URL: N/A
 License: N/A

Component Name	Pattern	Language	Repository URL	Homepage URL	License
ACPI	.*	C			
Block	.*block/.*	No	109,636	0.60	
Crypto	.*crypto/.*	No	72,817	0.78	
Arch-x86	.*arch/x86/.*	No	168,176	0.65	
Kernel	.*kernel/.*	No	154,864	0.77	
Lib	.*lib/.*	No	87,500	0.40	
MM	.*mm/.*	No	62,446	0.48	
KVM	.*kvm/.*	No	3,899	0.26	

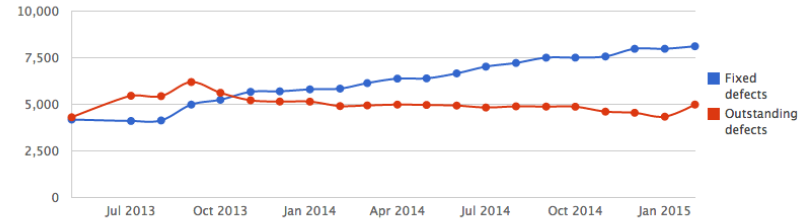
Version: 4.0.0-rc...

Feb 23, 2015 Last Analyzed	9,478,601 Lines of Code Analyzed	0.53 Defect Density
--------------------------------------	--	-------------------------------

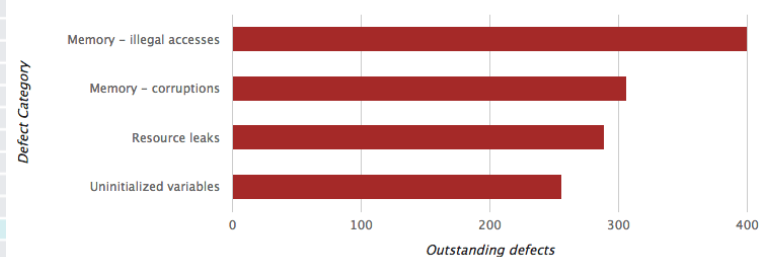
Defect changes since previous build dated Feb 09, 2015

244 Newly detected	227 Eliminated
------------------------------	--------------------------

Outstanding vs Fixed defects over period of time



High impact Outstanding Defect per Category



Establish an SDL for Software Security

- ◆ Provides process to follow and measure
- ◆ Supports resource and training requests
- ◆ Increases consistency across an organization
- ◆ Creates measures for reporting
- ◆ Initiates collection of best practices for your organization

◆ Microsoft Secure Development Lifecycle

The Microsoft Security Development Lifecycle						
Training	Requirements	Design	Implementation	Verification	Release	Response
Core Security Training	Establish Security Requirements	Establish Design Requirements	Use Approved Tools	Dynamic Analysis	Incident Response Plan	Execute Incident Response Plan
	Create Quality Gates / Bug Bars	Analyze Attack Surface	Deprecate Unsafe Functions	Fuzz Testing	Final Security Review	
	Security & Privacy Risk Assessment	Threat Modeling	Static Analysis	Attack Surface Review	Release Archive	

◆ Building Security In Maturity Model (BSIMM)

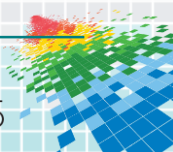
The BSIMM Software Security Framework (SSF)			
Governance	Intelligence	SSDL Touchpoints	Deployment
Strategy and Metrics	Attack Models	Architecture Analysis	Penetration Testing
Compliance and Policy	Security Features and Design	Code Review	Software Environment
Training	Standards and Requirements	Security Testing	Configuration Management and Vulnerability Management

Set Target Capability Maturity to balance risk and reward

- ◆ Common Criteria (& Orange Book/ Rainbow Series)
- ◆ Software Capability Maturity Model
- ◆ NIST Security Framework
- Apply as guides to your organizational needs

Sample Capabilities

Maturity Scale	High	<ul style="list-style-type: none">• Architecture driven S/W design• Comprehensive testing and review prior to operations• Fully isolated and segmented network(s)• Cleared Personnel
	Medium	<ul style="list-style-type: none">• Common Criteria – Higher Levels of Assurance throughout process• Strong BSIMM-like processes• SDL gate(s) prior to release
	Medium	<ul style="list-style-type: none">• Automation in component tracking and management• Supplier data sharing and collaboration• SDL Process in place• S/W sources minimally documented
	Low	<ul style="list-style-type: none">• Limited testing and SDL process controls• Software development process limited maturity• Minimal awareness of software sources
		<ul style="list-style-type: none">• No formal software cyber security controls• Software written ad hoc to meet functional requirements

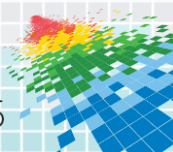


Review Code Repositories & Governance Processes

- ◆ **Who has access to your code? Directly and Indirectly?**
- ◆ **What protects your code? In Storage, Distribution, Use & Operation?**
- ◆ **Where is your Code stored? Used?**
- ◆ **When do you take control? When do you relinquish control?**
- ◆ **Why are you protecting it? What are the threats?**
- ◆ **How do you maintain integrity, authentication, and authorization?**

Core Principles

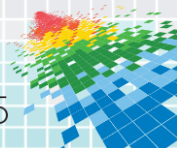
1. Confidentiality
2. Integrity
3. Accountability
4. Assurance



Conduct Supplier Software Assessment

- ◆ Assess each supplier from several aspects
- ◆ Meet with suppliers (where possible)
- ◆ Consider “Shared Assessments” Process
- ◆ Remember traditional supplier measures don’t apply to Open Source Developers who are:
 - ◆ Globally Distributed
 - ◆ Often known by “Handles”
 - ◆ Have unique, individual processes
 - ◆ May be ***security aware***
 - ◆ ***or functionality focused***

Example Aspects of Supplier Assessment



Supply Chain Software Assessment

Potential Risks and Recommendations for Cyber-related Software

Risk Rating	Risk Description	Recommendation
High	Lack of integrated cyber security testing across S/W components prevents understanding of vulnerabilities and attack surfaces	Require testing of integrated code prior to promoting code
Critical	Inconsistent dynamic testing (e.g. for backdoors that bypass cyber security controls) to prevent developer attacks and poor coding	Establish "Red Team" testing group focused on software security testing
High	Need for more consistent and comprehensive source code reviews to more effectively identify development-process vulnerabilities	Formalize static & dynamic code reviews with results reviewed prior to code approval
High	Testing of Open Source software often does not include cyber-related components	Review open source code with caution and test explicitly for cyber-related issues
Med	Ad hoc software modifications introduced late in the production process increasing potential exposure to malware	Document and enforce software Cyber Security policies and standards
Med	Minimal perspective on adversary supply chain operations limiting ability to anticipate FOSS cyber-related attacks	Enable appropriate Threat intelligence distribution to developers and testers

Common recommendations for managing your software supply chain #RSAC

Process

- Establish a risk based prioritization methodology
- Develop unique identifier to track critical s/w components
- Enhance contractual and tracking processes to provide control over origins of s/w including licenses and obligations
- Integrate application testing processes
- Develop robust remediation process

Skills & Training

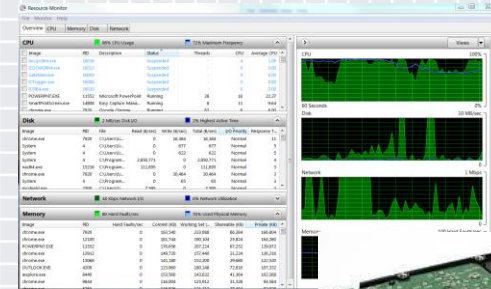
- Identify needed skills and training
- Develop an authoritative testing group
- Establish software security champions across areas
- Use communication campaigns to build culture of cyber security
- Use contract projects to learn from outside experts with deep knowledge

Technology

- Leverage software repositories to standardize code management – consolidate, integrate, and automate
- Use data analytics tools to provide component code visibility
- Use automated tools to integrate cyber testing for software
- Enhance identity and access controls for developers and suppliers

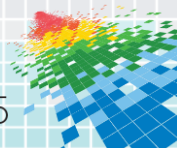
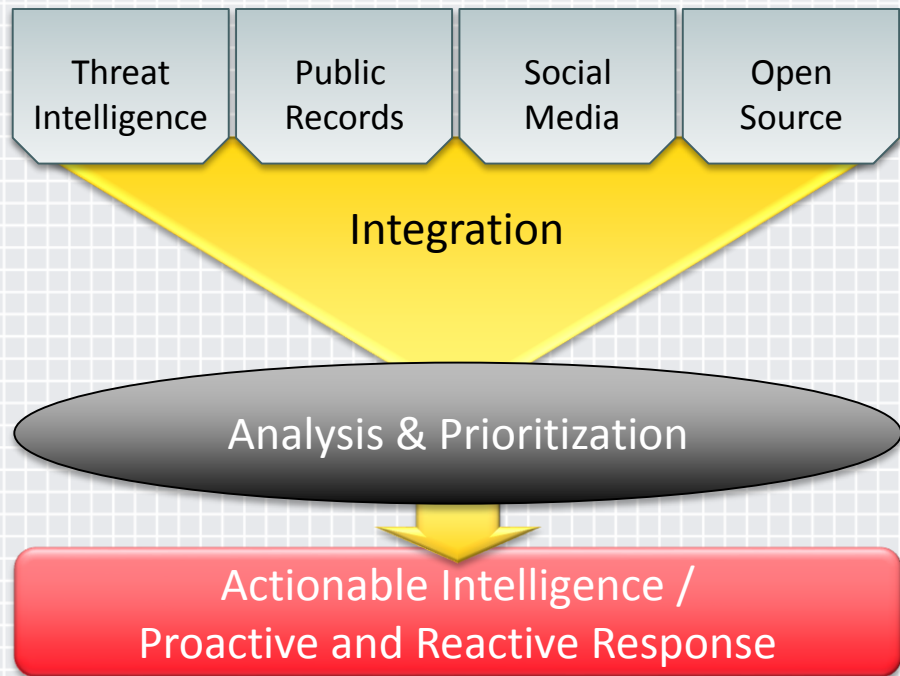
Implement Checks on Firmware (if your organization builds hardware)

- ◆ Operating Checks and Verification
- ◆ Trusted Boot Sequences
- ◆ Secure Elements in Hardware



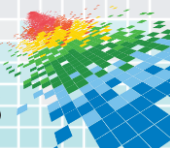
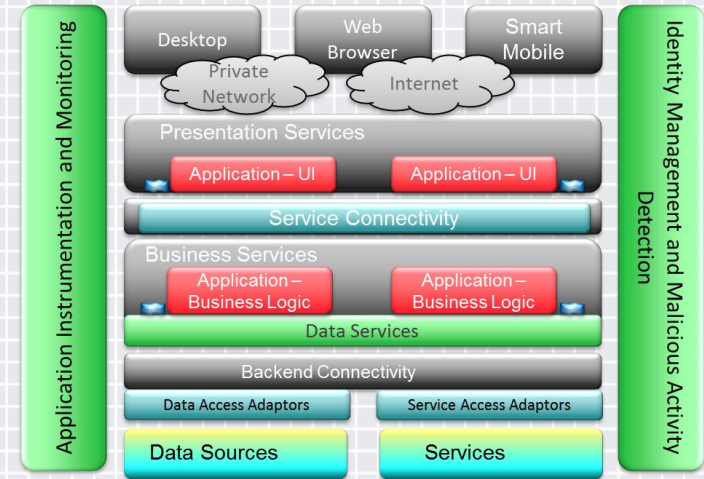
Monitor the changing Environment

- ◆ Threat Intelligence
- ◆ ISAC Participation
- ◆ Vendor Relationships
- ◆ Vulnerability Management
- ◆ Risk Management
- ◆ Incident Response

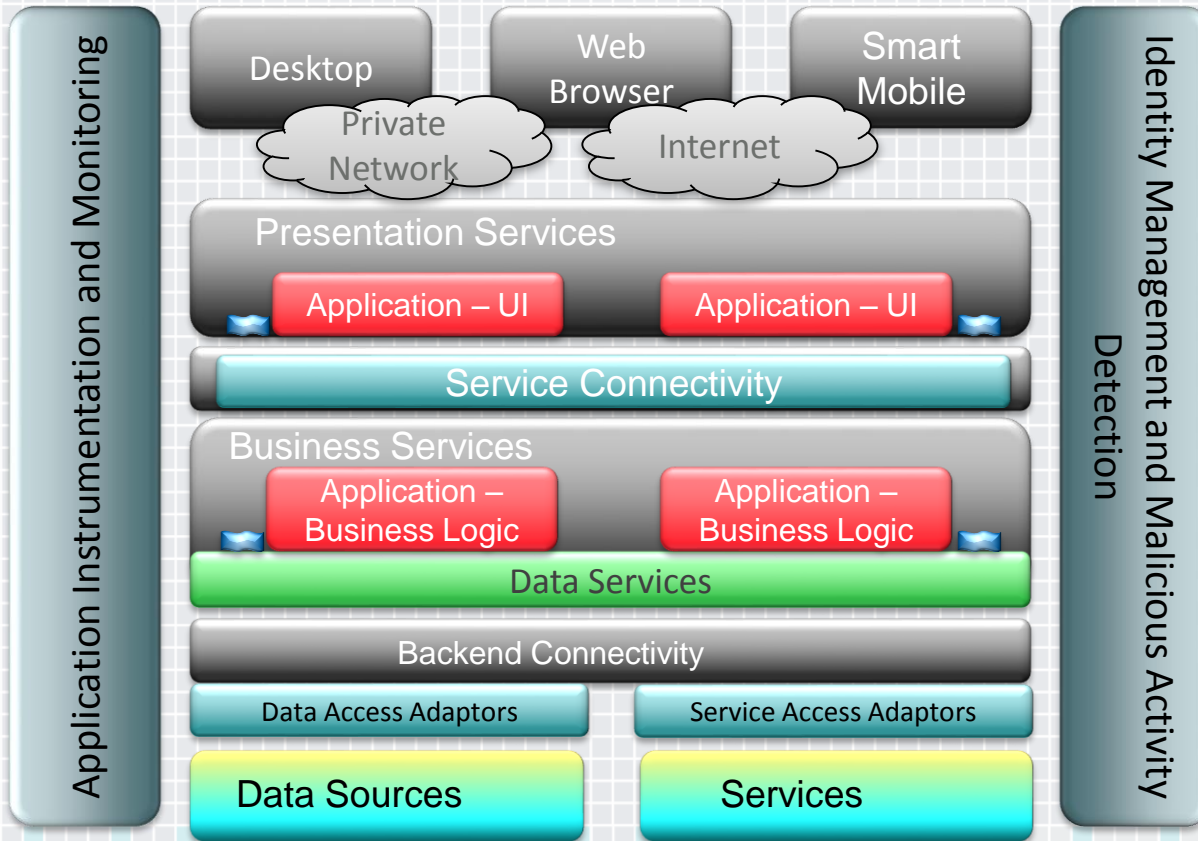


Design Resilient Systems

- ◆ Plan for systems to resist attack
- ◆ Reduce attack surfaces
- ◆ Eliminate excess
- ◆ Limit access directions
- ◆ Log and check logs

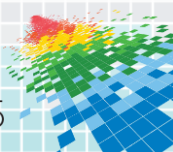


Design and architect systems to manage risk

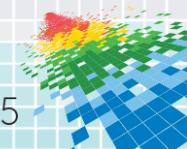
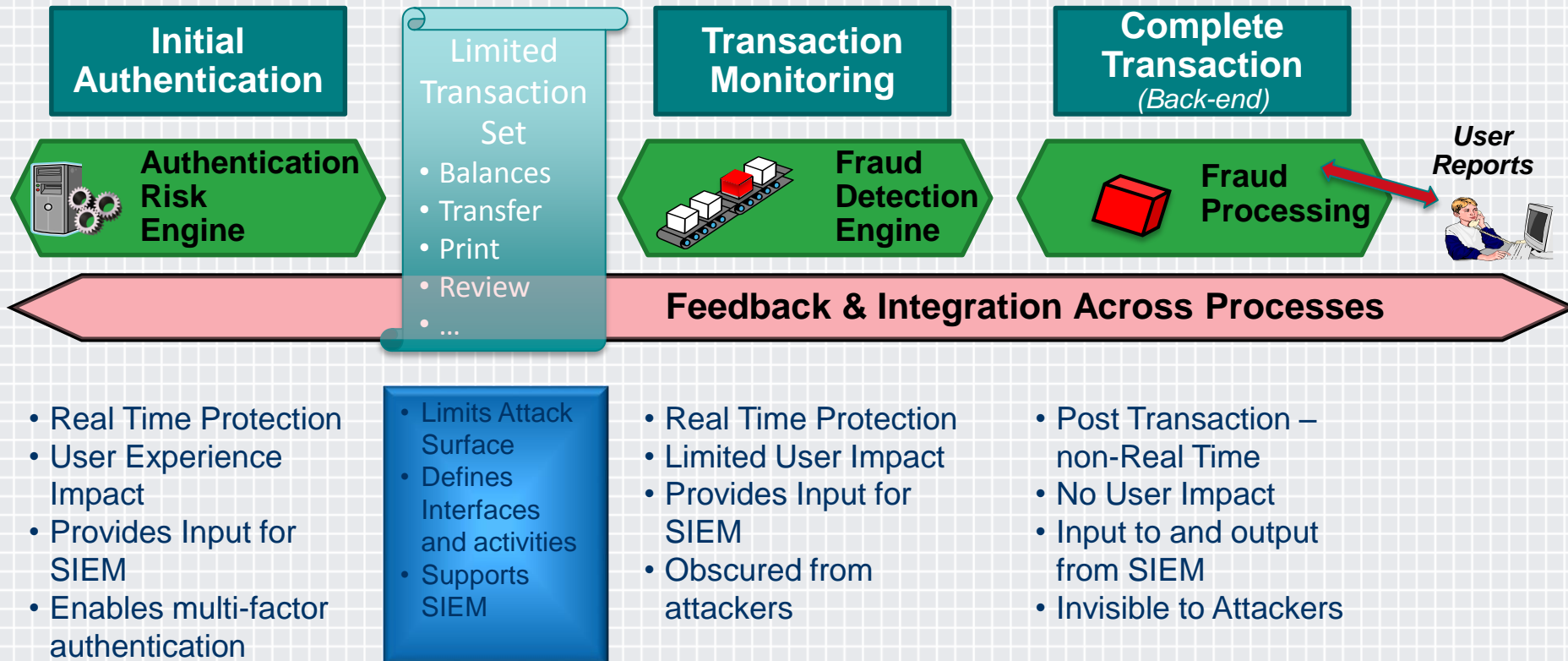


Key Elements

- Presentation services
- Service Connectivity
- Business Services
- Data Services
- Backend Connectivity
- Instrumentation and monitoring
- Identity Management
- Malicious Activity Detection

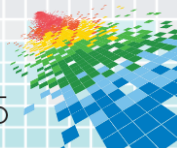


Worked Example – Online Banking Systems



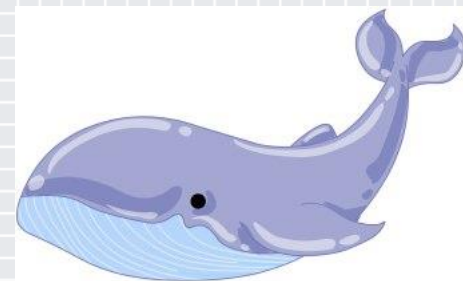
When you return – apply core principles

- ◆ **Establish** Ownership for Software Security
 - ◆ **Identify** Critical Software – Build a BOM
 - ◆ **Evaluate** FOSS & Licenses/Management
 - ◆ **Scan** Systems and Codebase
 - ◆ Compare Codebases to NVD
 - ◆ **Establish** SDLC for Software Security
 - ◆ **Consider** the Building Security In Maturity Model (BSIMM)
 - ◆ **Review** Code Repositories & Governance
 - ◆ **Conduct** Supplier Software Assessment
 - ◆ **Implement** Checks on Firmware
 - ◆ **Monitor** the changing Environment
- ◆ **Assess** What you Have
 - ◆ Ownership of Software Security?
 - ◆ Policies for Secure Software Development?
 - ◆ Tools for tracking and managing software?
 - ◆ Processes for implementing Tools & Policy?
 - ◆ Document gaps and needs
 - ◆ Build awareness and consensus for action
 - ◆ Find resources
 - ◆ Start simply

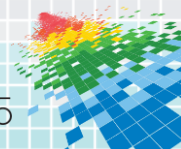


Asking, How Vulnerable is the Internet?

Turns out to be a bit like
asking for



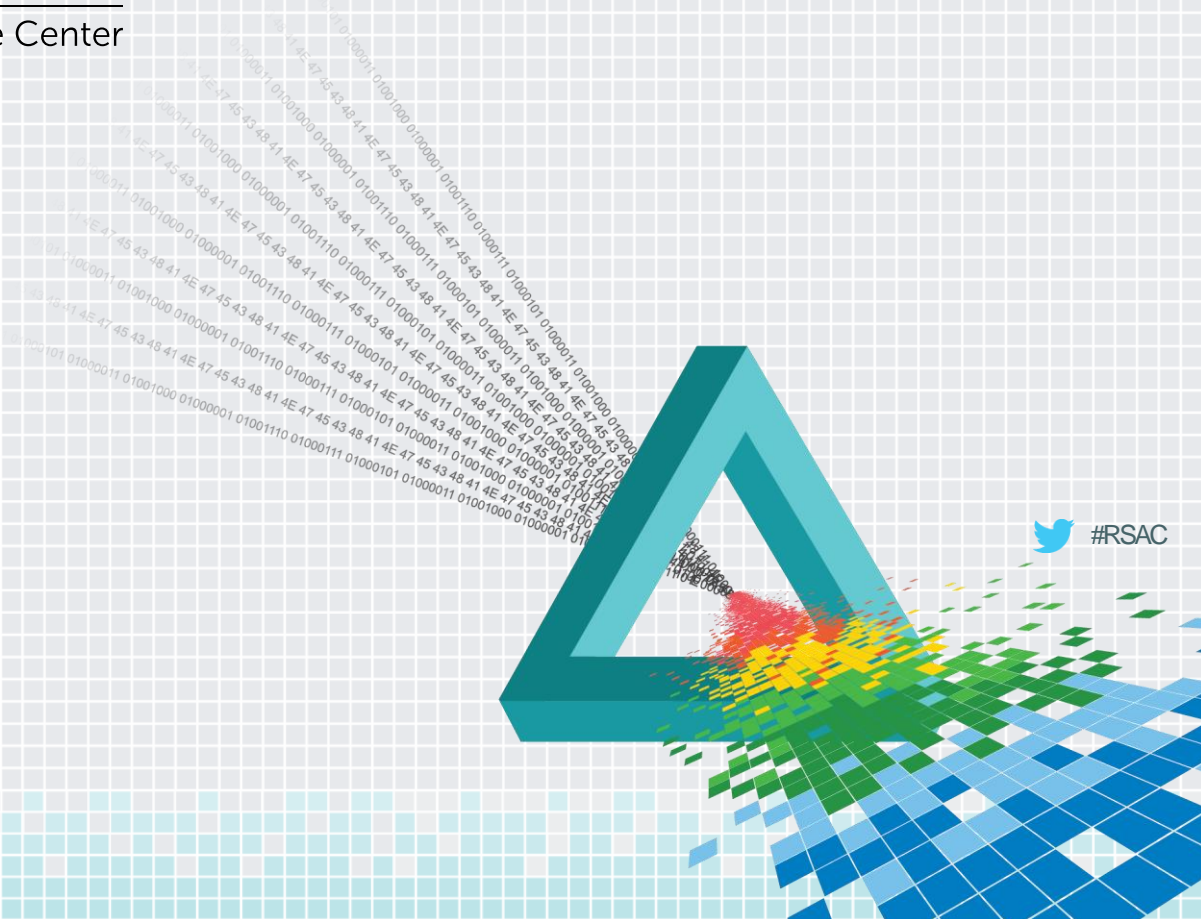
“The Answer to the Ultimate
Question of Life, the Universe , and
Everything”



RSA[®]Conference2015

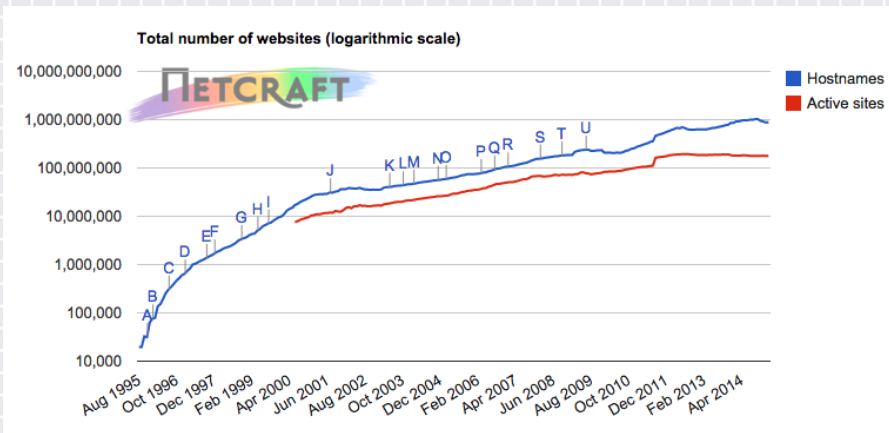
San Francisco | April 20-24 | Moscone Center

Backup / Afterthoughts



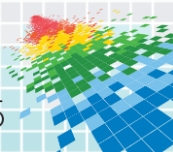
Why is Scanning the Internet so hard?

- ◆ The Internet is Colossally Huge
 - ◆ IPV4 – 4.3 **Billion** or 4.3×10^9 addresses
 - ◆ IPV6 – 3.4×10^{38} addresses



<http://news.netcraft.com/archives/2015/02/24/february-2015-web-server-survey.html>

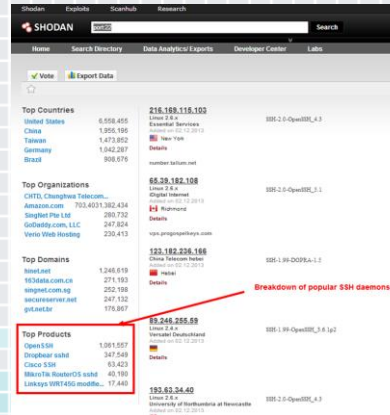
- ◆ Netcraft - February 2015 Survey
 - ◆ 883,419,935 sites / 5,135,229 web-facing computers
 - ◆ Microsoft IIS 28.7%
 - ◆ Apache 38.8%
- ◆ Versions?
- ◆ Underlying OS?
- ◆ Routers/Switches?
SCADA/PCS/ICS
IoT – Tesla’s, Nest’s, etc
- ◆ How old are these things?



Scan the Internet

- ◆ Several groups are scanning aspects of the Internet
 - ◆ University of Michigan, Netcraft, Electronic Frontier Foundation and Insecure.org, and individuals
- ◆ Variety of Tools Available
 - ◆ Scan.IO (<https://scans.io/>)
 - ◆ Shodan(<http://www.shodanhq.com/>)
 - ◆ NMAP(<http://nmap.org/>)
 - ◆ Masscan(<https://github.com/robertdavidgraham/masscan>)
 - ◆ ScanHub(<https://scanhub.shodan.io/>)
 - ◆ Zmap(<https://zmap.io/>)
- ◆ Powerful hardware and network connection
- ◆ Excellent Exclusion List
- ◆ Pre-arrange with ISP

→ Hard and Getting Harder to really understand the Internet



The screenshot shows the Shodan search engine interface. The search results are categorized into several sections:

- Top Countries:** United States (6,558,405), China (1,956,195), Taiwan (1,472,802), Germany (1,042,267), Brazil (906,676).
- Top Organizations:** CHTI, Changchun Telecom... (66,38,182,108), Amazon.com (702,4031,392,434), Spigdetline Ltd (280,720), GoDaddy.com, LLC (247,824), Verio Web Hosting (230,413).
- Top Domains:** intel.net (1,246,619), 1000000.com.au (276,193), usmapnet.com.sg (252,199), saracenter.net (247,132), gfw.taipei (175,967).
- Top Products:** OpenSSH (1,081,057), (Shodan) sshd (347,543), Cisco SSH (83,423), MikroTik RouterOS sshd (43,106), Linksys VRS145G module... (17,443).

The main search results list includes:

- SSH-2.0-OpenSSH_4.3 (116,159,115,103)
- SSH-2.0-OpenSSH_3.1 (66,38,182,108)
- SSH-1.99-OpenSSH_1.1 (121,182,208,188)
- SSH-2.0-OpenSSH_1.8.1p2 (67,246,255,59)
- SSH-2.0-OpenSSH_4.3 (193,62,34,42)

```
nmme/jose # zmap -p 443 -o results.txt
49,354 [INFO] zmap: started
id: 139718 140 Kp/s (138 Kp/s avg); recv: 20 19 p/s (19 p/s avg); dr
p/s avg); hits: 0.01%
id: 282771 143 Kp/s (140 Kp/s avg); recv: 47 26 p/s (23 p/s avg); dr
p/s avg); hits: 0.02%
id: 424889 142 Kp/s (141 Kp/s avg); recv: 71 23 p/s (23 p/s avg); dr
p/s avg); hits: 0.02%
id: 552626 128 Kp/s (137 Kp/s avg); recv: 97 25 p/s (24 p/s avg); dr
p/s avg); hits: 0.02%
5m left); send: 696299 144 Kp/s (139 Kp/s avg); recv: 130 32 p/s (2
drops: 0 p/s (0 p/s avg); hits: 0.02%
2m left); send: 839796 143 Kp/s (139 Kp/s avg); recv: 155 24 p/s (2
drops: 0 p/s (0 p/s avg); hits: 0.02%
3m left); send: 984162 144 Kp/s (140 Kp/s avg); recv: 184 28 p/s (2
drops: 0 p/s (0 p/s avg); hits: 0.02%
9m left); send: 1127676 144 Kp/s (141 Kp/s avg); recv: 217 32 p/s (
drops: 0 p/s (0 p/s avg); hits: 0.02%
```

Scanning the Internet

- ◆ The Internet-Wide Scan Data Repository
 - ◆ public archive of research data
 - ◆ active scans of the public Internet.
 - ◆ hosted by the [ZMap Team](#) at the [University of Michigan](#).
 - ◆ Hosts scan data from others
 - ◆ A [JSON interface](#) to the repository is also available.

Internet-Wide Scan Data Repository

The Internet-Wide Scan Data Repository is a public archive of research data collected through active scans of the public Internet. The repository is hosted by the ZMap Team at the University of Michigan. While the ZMap team publishes much of the data, we are happy to host scan data from other researchers as well. Please contact Zakir Durumeric with any questions. A JSON interface to the repository is also available.

University of Michigan - Daily Full IPv4 HTTPS Handshakes

Daily ZMap scans of TCP/443 and parsed TLS handshakes with responding hosts.

University of Michigan - Daily Full IPv4 Modbus MEI-DEVICE-ID

Daily ZMap scans of TCP/502 and self-reported device information.

University of Michigan - Weekly IPv4 HTTPS Heartbleed

Daily ZMap scans of TCP/443 and heartbleed vulnerability check.

University of Michigan - Daily Full IPv4 FTP Banner Grab

Daily ZMap scans of TCP/21 and ZGrab Banner Grab with responding hosts.

University of Michigan - Daily Full IPv4 CWMP GET /

Daily ZMap scans of TCP/7547 and ZGrab GET / from responding hosts

University of Michigan - Daily Alexa Top 1 Million HTTPS Handshakes

Daily HTTPS ZGrab scan of the Alexa Top 1 Million domains.

University of Michigan - Weekly IPv4 SSH (RSA) Banner Grab

Weekly ZMap scan of IPv4 targeting TCP/22 and Banner Grab of RSA key.

Netcraft

- ◆ Offers a variety of pay services related to Internet presence
- ◆ Hidden Gem with regular public updates
- ◆ Provides reality check against Forrester, Gartner, and IDC predictions



NETCRAFT

Home News Anti-Phishing Security Testing Internet Data Mining Performance About Netcraft

Internet Security and Data Mining

Netcraft provide internet security services including [anti-fraud and anti-phishing services](#), [application testing](#) and [PCI scanning](#). We also analyse many aspects of the internet, including the [market share of web servers](#), [operating systems](#), [hosting providers](#) and [SSL certificate authorities](#).

Anti-Phishing Security Testing **Internet Data Mining** Performance

Market Share for Top Servers Across All Domains



Year	Apache	Microsoft	Sun	nginx	Google	NCSA	Other
2000	75%	25%	5%	0%	0%	0%	0%
2005	65%	35%	5%	0%	0%	0%	0%
2010	60%	35%	5%	0%	0%	0%	0%
2015	65%	25%	5%	15%	0%	0%	0%

Understand your Competitors

- ▶ Worldwide analysis of hosting companies, identifying trends and customer movements
- ▶ Track technology adoption across the internet including the market share of web servers, operating systems, hosting providers and SSL certificate authorities
- ▶ See a list of all websites that match requested criteria (for example sites running a certain technology hosted in a particular country)
- ▶ [Find out more](#)

