Universität Hamburg

Exploiting the Intranet With a Webpage

Is JavaScript the New Shellcode?

HITB SecConf 2007

05. September 2007

Martin Johns

Fachbereich Informatik
SVS – Sicherheit in Verteilten Systemen

Me, myself and I

Martin Johns

- johns at informatik.uni-hamburg.de
- Security researcher at the University of Hamburg
- Member of the secologic project
  - Research project carried out by SAP, Commerzbank, Eurosec and the University of Hamburg
  - Sponsored by the German Ministry of Technology (BMWi)
  - Goal: Improving software security
  - Visit us at http://www.secologic.org

JavaScript Malware?

Term coined by Jeremiah Grossman

Describes a class of browser-based-attacks that target intranet resources

A lot of ongoing research since late 2006

All attacks covered in this talk are “legal” actions according to the HTTP, HTML and JavaScript specs/drafts/RFCs

No browser bugs required

Agenda

- The Basics
- Intranet Attacks
- DNS Rebinding
- Client Side Protection
- Conclusion

Agenda

- **The Basics**
- Intranet Attacks
- DNS Rebinding
- Client Side Protection
- Conclusion

The Same Origin Policy (SOP)

Designed to prevent cross-domain read/write access

- Applies to JavaScript
- Affects cookie-access, cross-document interaction and networking communication
- The SOP is satisfied iff
  - the protocol,
  - the domain and
  - the port

  of two elements match
- Java and Flash have similar policies

The Same-Origin Policy (II)

So, the SOP provides a nice sandbox:

1. **No direct access to the local file system (Protocol-rule)**
2. **No direct access to other hosts (Domain-rule)**
3. **No direct access to other applications on the same host (Port- and protocol-rule)**

A loophole in the SOP

JavaScript can initiate network communication through dynamic inclusion of elements in the DOM-tree:

- The script includes a HTML element in the page which references a network resource
  - IMG, STYLE, SCRIPT, IFRAME
- By this inclusion of such an element the browser creates an HTTP request

The targets of such requests are not restricted by the SOP

- This in fact enables *indirect* cross-domain communication
- Outgoing data:
  - URL parameters
- Incoming data:
  - Side effects of the inclusion process

Circumventing the SOP (I)

Remember: The SOP prevents cross-domain data-retrieval

- Does it?

The basic reconnaissance attack (BRA)

- Question: Does the element with URL U exist?

Method:

- Construct URL U pointing to the target of the examination
- Start a timeout-event t
- Include a suiting network aware element using U
- Use JavaScript's eventhandler-framework to determine the result:
  - The timeout t occurs ⇒ The target does not exist
  - onload() event ⇒ The target exists
  - onerror() event ⇒ (specific result depends on the element and target - stay tuned)

Circumventing the SOP (II)

Remember: The SOP prevents cross-domain write access

- Does it?

Cross Site Request Forgery

- CSRF aka XSRF
- aka Session Riding
- aka Sea Surf

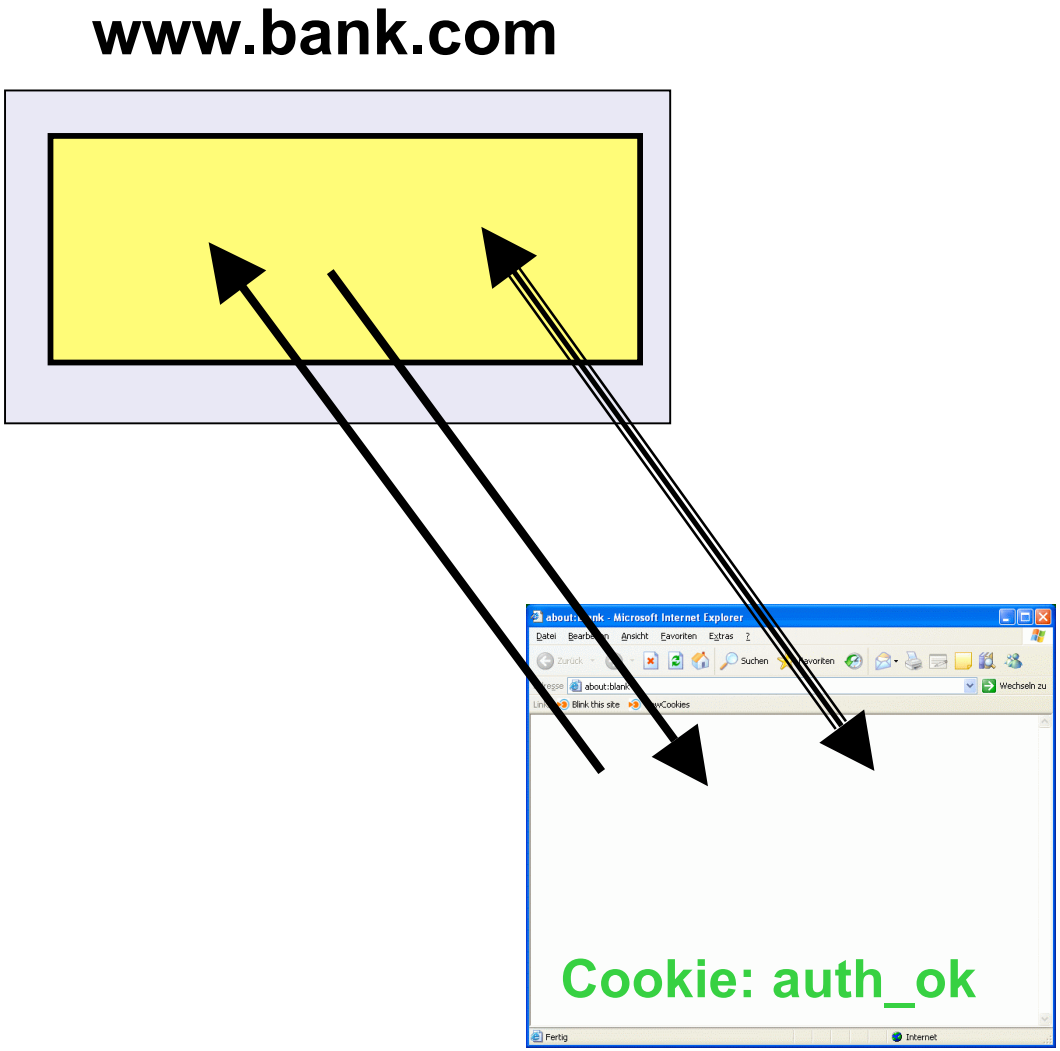
Implicit authentication

- Auth. mechanisms that are executed by the browser without user interaction
- Cookies, HTTP Auth, NTLM, client-side SSL

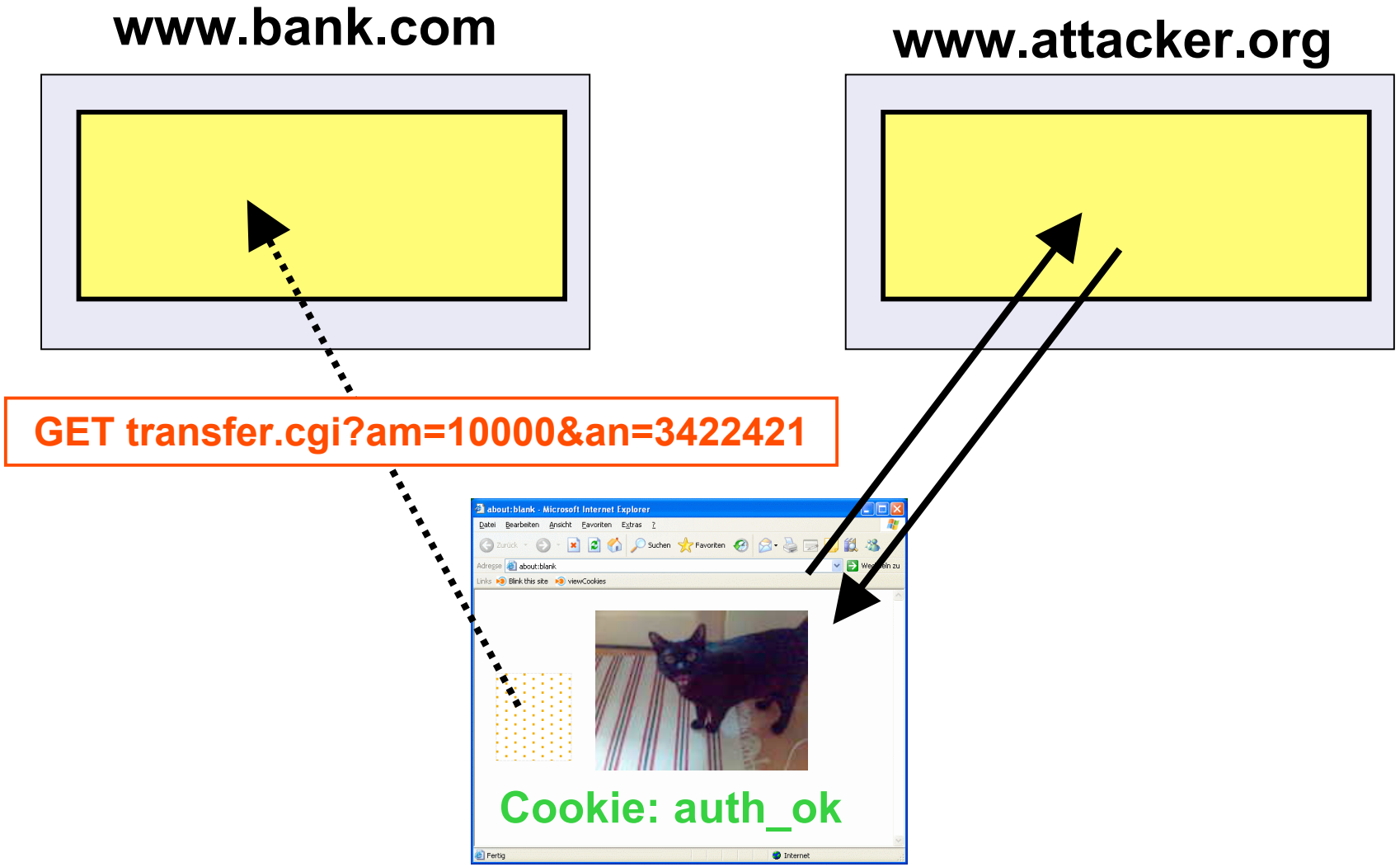
CSRF exploits implicit authentication mechanisms

- Creation of hidden, state-changing cross-domain requests
- These requests are automatically outfitted with the user's credentials

Cross Site Request Forgery (CSRF)



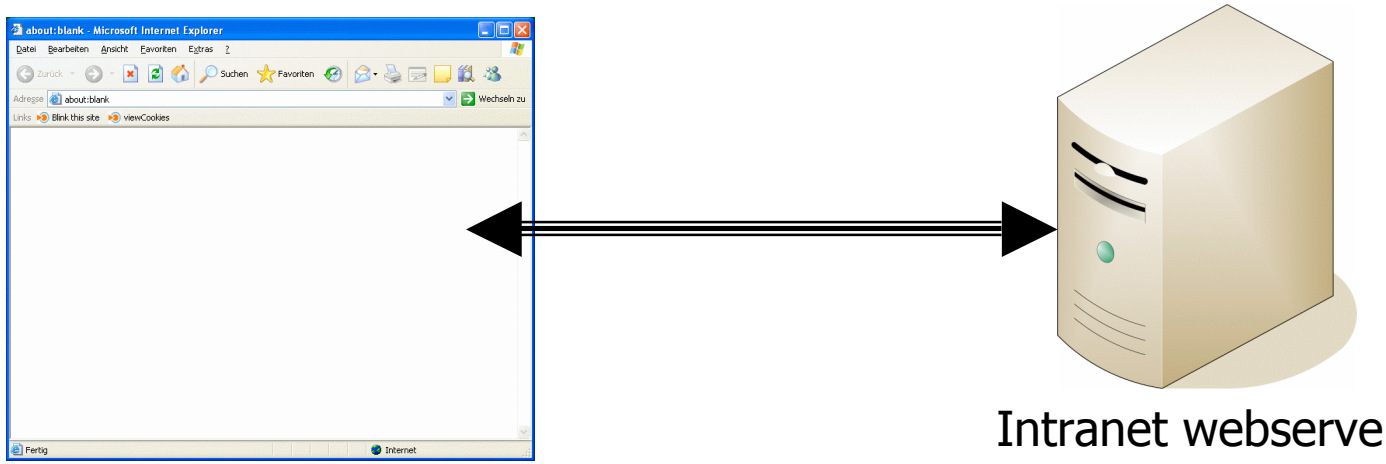
Cross Site Request Forgery (CSRF)



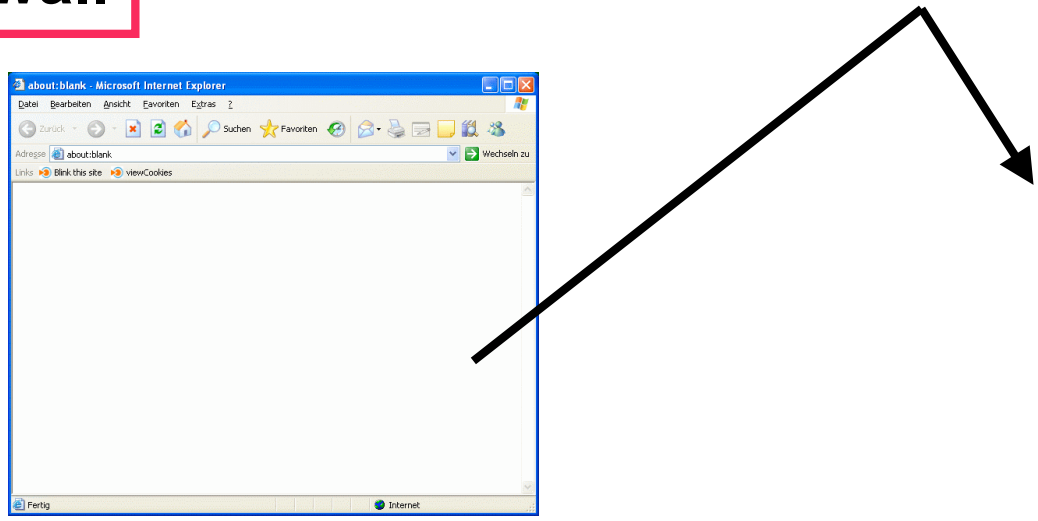
Agenda

- The Basics
- **Intranet Attacks**
- DNS Rebinding
- Client Side Protection
- Conclusion

IP based authentication



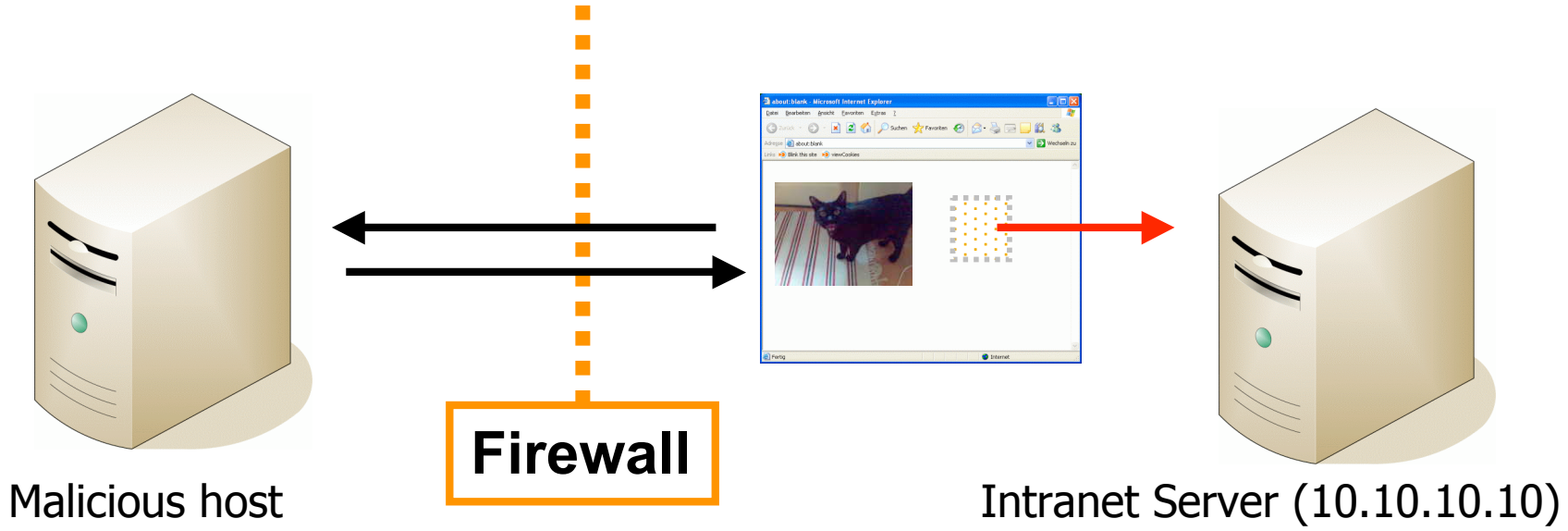
Firewall



Subverting the Firewall with CSRF

Firewall == implicit mean of authentication

⇒ **Susceptible to CSRF**



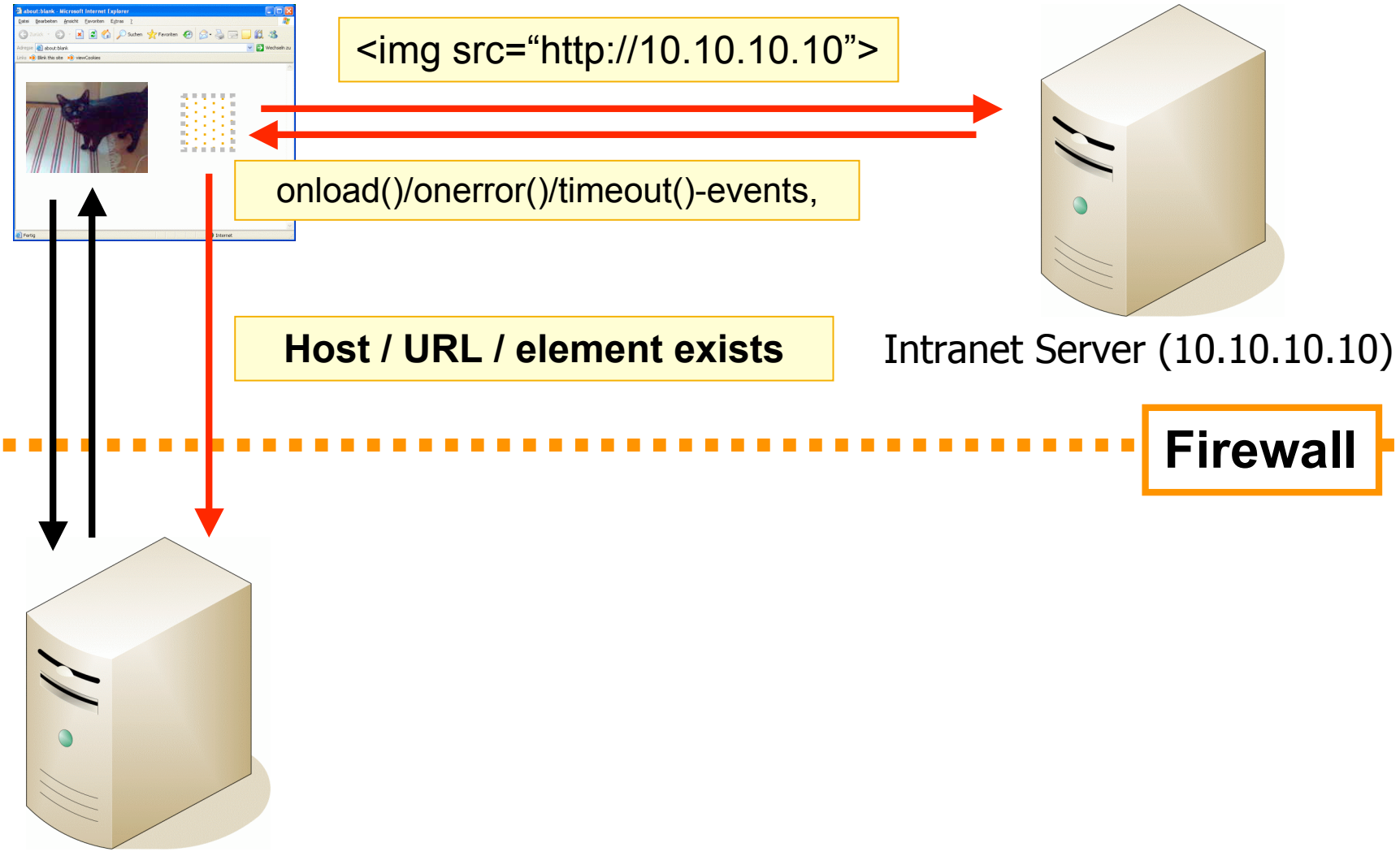
Putting it all together

By looking at a webpage we allow JavaScript-execution within the intranet...

As we have just discussed, JavaScript can do

- Reconnaisance (BRA)
- Exploting (CSRF)

Intranet exploration with the BRA



But where to start?

“My hosts are NATed and use obscure private IPs”

Java to the rescue:

- Java applets provide low level sockets
- The target of these sockets is restricted by the SOP
- This does not matter as we are interested in the origin of the connection (the local IP)
- On modern browsers even more convenient with LiveConnect:

```
function natIP() {
  var w = window.location;
  var host = w.host;
  var port = w.port || 80;
  var Socket = (new java.net.Socket(host,port))
                    .getLocalAddress().getHostAddress();
  return Socket;
}
```

Intranet exploration with the BRA (II)

Ping sweep / http-server discovery:

- Iterate through the subnet using the BRA
- <IFRAME src="http://10.10.10.1">,
  <IFRAME src="http://10.10.10.2">,
  <IFRAME src="http://10.10.10.3">,
  ...
  - Timeout-event: Host does not exist
  - OnLoad-event: Host runs a webserver
  - OnError-event: Host exists but the port is closed (RST package)
- Varying the port might locate https or development servers

Intranet exploration with the BRA (III)

Server/application discovery/fingerprinting

- Known “special” DNS names
  - <IFRAME src=“http://fritz.box”> (home router)
- Known image-URLs
  - <IMG src=“http://10.10.10.10/icons/c.gif”> (Apache)
- Web page fingerprinting based on JavaScript errors
  - <SCRIPT src=“http://10.10.10.10/index.php”>

```
<script>
function err(msg, url, code) {
   if ((msg == "missing } in XML expression" ) && (code == 1)) {
       // Wordpress
   } else if ((msg == "syntax error" ) && (code == 3)) {
       // Squirrelmail
   } else
       // unknown
}

window.onerror = err;
</script>
```

Intranet exploration with the BRA (IV)

HTTP-authentication

- If the scanned server is protected by HTTP-auth the browsers displays a login-dialogue
- This should at least startle the browser's user

Avoiding HTTP-authentication pop-ups (Stefan Esser)

- The trick is to cause the server to drop the request before it is processed
- This can be achieved by malformed URLs
- Incomplete entities:
  - http://host/%
- Excessively long URLs
  - http://host/AAA ... AAA
- Breaks fingerprinting

Exploiting the intranet

The attacker is able to:

- locate intranet hosts and
- fingerprint applications/routers/devices

Several promising points for CSRF attacks:

- Unchanged default passwords on appliances
  - “Drive by Pharming”
- Unpatched servers
  - The old and almost forgotten IIS in the basement
- Outdated intranet applications
  - Wordpress 2.0 for internal communication

Some limitations...

Timing

- Working with timeout-events takes... Time
- Using parallelization can speed the process up
- But various restrictions on connections limits exit
  - Windows XPSP2 and later

Port restrictions

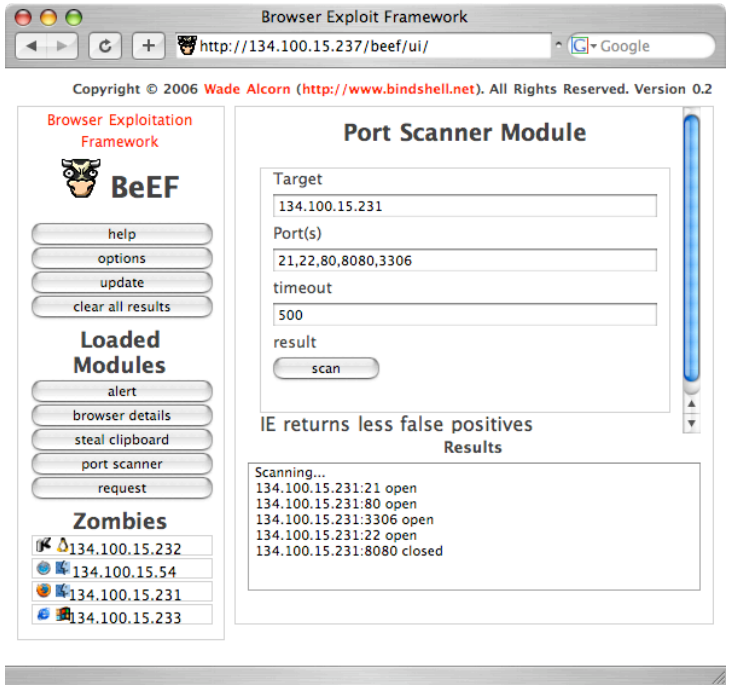
- Most browsers only allow HTTP and high-number ports

Further attacks

- Fingerprinting / attacking non http-protocols via multi-part HTML forms
- Attacks that don't require JavaScript

Convenient attack tools exist

- E.g., Browser Exploitation Framework (BeEF)
  - One line XSS-payload
- Backframe



Agenda

- The Basics
- Intranet Attacks
- **DNS Rebinding**
- Client Side Protection
- Conclusion

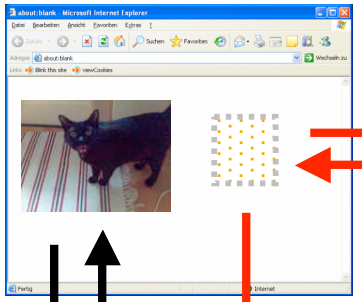
DNS spoofing / anti-DNS-pinning / DNS-rebinding

Originally invented 1996 to subvert Java applets

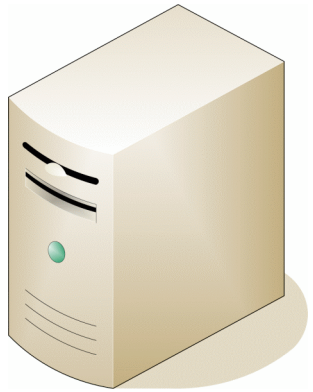
General technique

- The attacker dynamically created DNS entries assigned to local IP addresses
- This way the SOP can be circumvented

DNS spoofing / anti-DNS-pinning / DNS rebinding



http://attacker.org/index.html

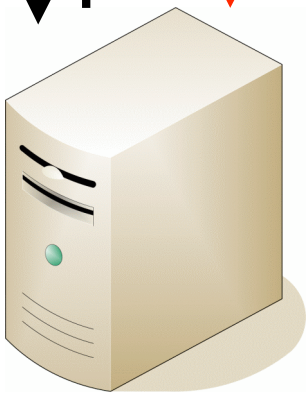


http://attacker.org/foo.html

Contents of the server

10.10.10.10 = attacker.org

Firewall



attacker.org == attacker.org
⇒ The SOP is safisfied

200.200.200.200 = attacker.org

DNS pinning

Counter Measure

- Keep the DNS binding for the livetime of the browser session
- Breaks, e.g., dynamic DNS, certain load balancing techniques
- Further problem: nowadays our browser sessions are quite long
- Violates RFC 2616

“Anti-DNS-pinning”

- In Firefox, IE and Opera not fully implemented
  - Issue open since approx. one year
  - Unknown if and how it will be fixed
- Methods to cause the browser to drop the pinning
  - Close the original port on attacker.org
  - Request a resource on a closed port on attacker.org
- Browsers take different amount of time to drop the pin
  - IE is the fastest

Countering DNS rebinding attacks

Host header:

- All requests created through JavaScript are within the domain “attacker.org”
  - Dictated by the SOP

⇒ Host-header == “attacker.org”

⇒ Web content for other virtual hosts is unreachable for JavaScript

“Anti-Anti-Anti-DNS pinning”

- It used to be possible to forge the host header:
  - with XMLHttprequest
  - with Flash
- Both vectors are fixed and work only on outdated browsers

DNS rebinding with LiveConnect

Remember LiveConnect?

- JavaScript can dynamically create Java objects

```
function natIP() {
  var w = window.location;
  var host = w.host;
  var port = w.port || 80;
  var Socket = (new java.net.Socket(host,port))
                    .getLocalAddress().getHostAddress();
  return Socket;
}
```

- TCP sockets
- What happens if such an object is created AFTER the DNS entry has changed?

⇒ The Java-runtime has its own pinning table

⇒ Java only sees the changed mapping

⇒ TCP sockets can be used in the attack

DNS rebinding with Flash

Flash 9 also has TCP sockets

- ...what Flash does not provide (yet) is DNS Pinning

⇒ Flash also oufits the attacker with TCP socket-coonection to intranet hosts

- Rebinding attacks are quite fast, determined by the entries TTL

DNS rebinding and TCP sockets

TCP sockets enable the attacker to recreate HTTP

- This implies creation of arbitrary HTTP headers

⇒ The host-header is useless, *again*.

Recreation of HTTP, part II

- 401 responses don't cause pop-ups anymore

⇒ password brute-forcing

Further capabilities through TCP sockets:

- Other protocols
- Binary data
- Basically, everything

Check out Billy K. Rios talk tomorrow!!!!

Agenda

- The Basics
- Intranet Attacks
- DNS Rebinding
- **Client Side Protection**
- Conclusion

Client Side Protection

Why at the client side?

- The server receives correct http requests from valid intranet hosts
- Server side indicators:
  - External referrer header
  - Mismatching host header, in the case of DNS-based attacks
- Both indicators can be evaded
  - Referrer headers can be deterministically suppressed
  - Host headers can be spoofed

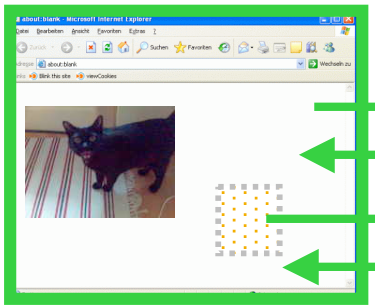
LocalRodeo

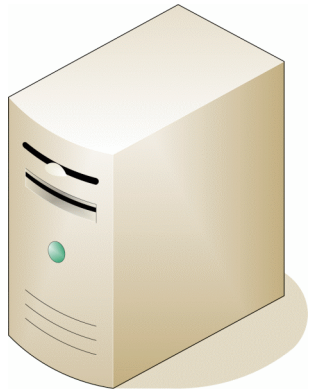
Concept: Segmentation based on the origin of webpages

- Local pages:
  - Retrieved from intranet locations
- Remote pages:
  - Retrieved from outside locations
- Classification is based on IP-address-ranges

Only local pages (== resources that have a local origin) are allowed to create requests to intranet locations

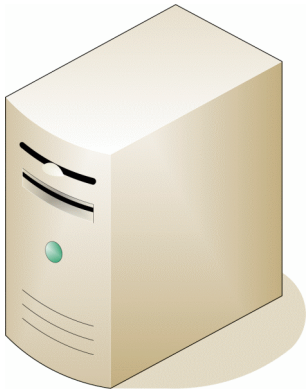
LocalRodeo (II)





Intranet Server (10.10.10.10)

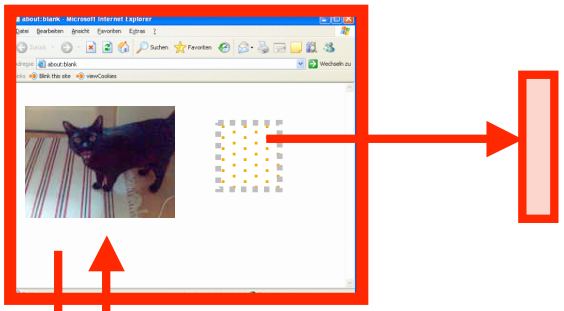
Firewall

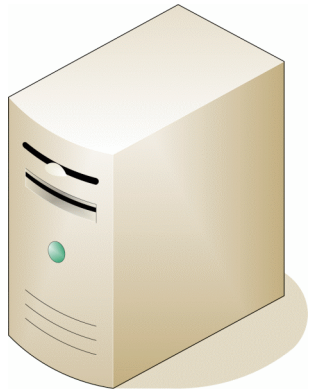


Malicious host

Webpage is tagged as “local”

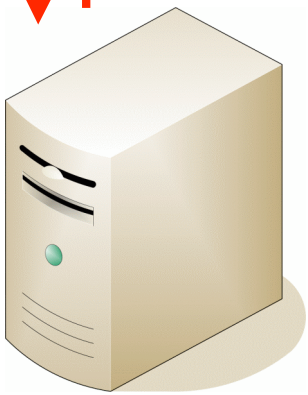
LocalRodeo (III)





Intranet Server (10.10.10.10)

Firewall



Malicious host

Webpage is tagged as “remote”

LocalRodeo (IV)

DNS rebinding

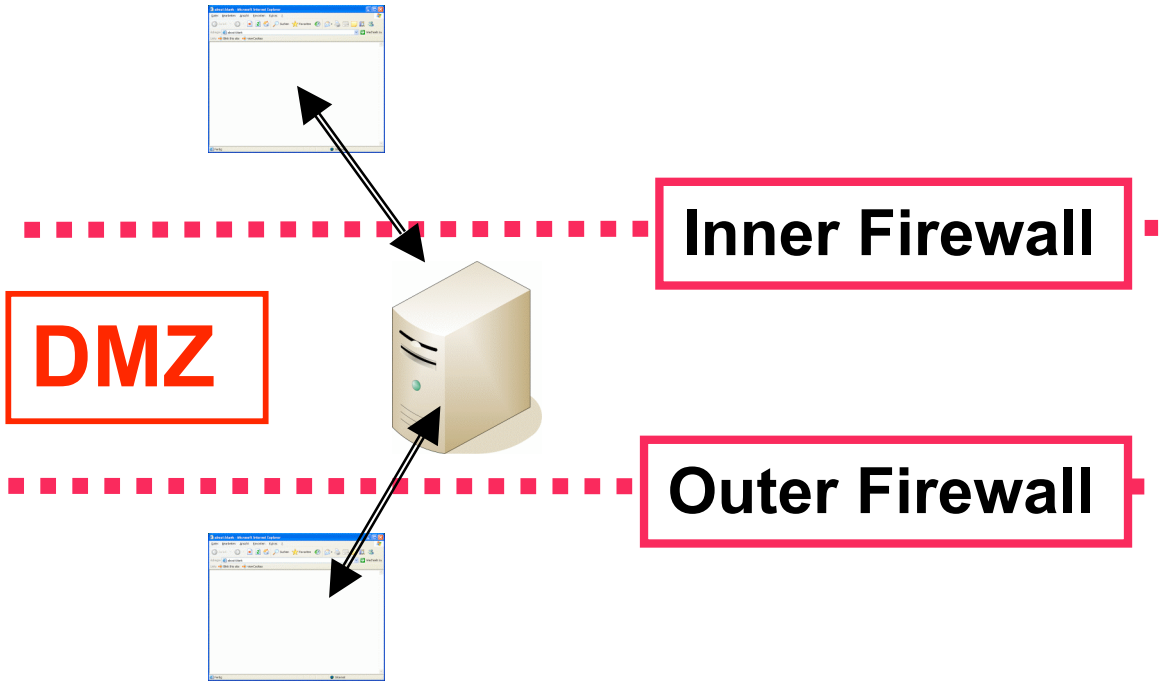
- LocalRodeo classifies into local/remote

Rebinding attack == switch from “remote” to “local”

- Rather easy to spot and stop

Limitation:

- A local/remote classification is not in all cases possible
- Example:
  - DMZ resources that grant different access rights based on source IP address



LocalRodeo (VI)

Advantages

- Good protection against all specified attacks
- Easy configuration for “simple” networks

Disadvantages

- Complicated configuration for “sophisticated” networks
  - How should different network segments be treated?
  - E.g., protecting against attacks from the inside of the same company
- No protection against Java and Flash based attacks

Implementation

Extension for the Firefox browser

Get it: http://databasement.net/labs/localrodeo

Still in beta

- if you would like to contribute, go ahead, it's open source :-)

The NoScript-developers have announced to include the proposed techniques in a future release

Countering DNS rebinding

Stanford researchers also investigated DNS rebinding

- They proposed / implemented a couple excellent countermeasures
- Firewall
  - Monitors DNS traffic
  - Denies external hostnames to resolve to internal IP addresses
- Check it out: http://crypto.stanford.edu/dns/

Keep in mind:

- Does only protect against rebinding attacks
- BRA and CSRF are still possible

Agenda

- The Basics
- Intranet Attacks
- DNS Rebinding
- Client Side Protection
- **Conclusion**

What did I not tell you?

Attacks that do not rely on JavaScript

- CSS based ping-sweeping
- External timing-based reconnaissance attacks

Privacy attacks

- Browser History-Disclosure
- Local machine profiling
- Timing attacks

Using the browser as an attack proxy

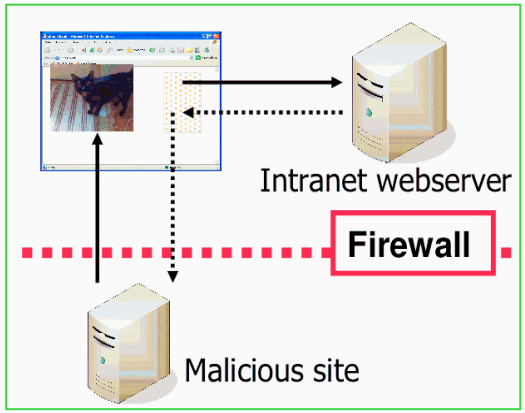
- Click-Fraud
- Server scanning (Nikto)
- Helping worm propagation (puppetnets)

...check out the bibliography when you have time

Summary

A rogue webpage can:

- Obtain the (internal) IP address of the hosting web browser
  - Using Java or guessing based on other evidence (existing URLs)
- Portscan the LAN to locate intranet http servers
  - Using the BRA while suppressing HTTP auth dialogues
- Fingerprint these http servers using well known URLs
- (sometimes) exploiting them via CSRF or
  - In case the fingerprinting found a known and vulnerable application
- Access the servers content and leak it to the outside by breaking DNS pinning
  - And use sockets for more sophisticated attacks



Think: XSS payload

- Remember the Samy worm

Conclusion / Lessons learned / Outlook

Intranet Servers

- Do not solely rely on the firewall to protect sensitive intranet services
- Apply additional explicit authentication
- Do not leave intranet servers unpatched

Client Side

- Disable Flash!
- Disable Java
- Use NoScript (and/or LocalRodeo)

Conclusion

- The SOP is insufficient
- ... and so is DNS pinning
- Relying on DNS for security purpose is not a good idea, anyway
  - DNS is not controlled by the web application
- Soundly solving the problem exclusively on the client-side (i.e., in the browser) is not feasible
  - At least as long certain cross-domain requests are permitted
- Future work should investigate server based policies concerning cross-domain interaction

The end

Thank you for your attention

Questions?

Comments?

The bibliography

[1] Wade Alcorn. Inter-protocol communication. Whitepaper, http://www.ngssoftware.com/research/papers/InterProtocolCommunication.pdf, (11/13/06), August 2006.

[2] Andrew Bortz, Dan Boneh, and Palash Nandy. Exposing private information by timing web applica- tions. In WWW 2007, 2007.

[3] Jesse Burns. Cross site reference forgery - an introduction to a common web application weakness. Whitepaper, https://www.isecpartners.com/documents/XSRF Paper.pdf, 2005.

[4] David Byrne. Anti-dns pinning and java applets. Posting to the Bugtraq mailing list, http://seclists.org/fulldisclosure/2007/Jul/0159.html, July 2007.

[5] David Byrne. Intranet invasion through anti-dns pinning. Talk at the Black Hat 2007 conference, August 2007.

[6] Mozilla Developer Center. Liveconnect. [online], http://developer.mozilla.org/en/docs/LiveConnect, (08/08/07), 2007.

[7] Andrew Clover. Css visited pages disclosure. Posting to the Bugtraq mailing list, http://seclists.org/bugtraq/2002/Feb/0271.html, February 2002.

[8] Thai N. Duong. Zombilizing the browser via flash player 9. talk at the VNSecurity 2007 conference, http://vnhacker.blogspot.com/2007/08/ zombilizing-web-browsers-via-flash.html, August 2007.

[9] Stefan Esser. Javascript/html portscanning and http auth. [online], http://blog. php-security.org/archives/54-JavaScriptHTML-Portscanning-and-HTTP-Auth. html, (08/27/07), November 2006.

The bibliography

[10] Edward W. Felten and Michael A. Schneider. Timing attacks on web privacy. In Proceedings of the 9th ACM Conference on Computer and Communication Security (CCS]02), 2000.

[11] Eric Glass. The ntlm authentication protocol. [online], http://davenport.sourceforge.net/ntlm.html, (03/13/06), 2003.

[12] Jeremiah Grossman. Browser port scanning without javascript. [online], http://jeremiahgrossman.blogspot.com/2006/11/ browser-port-scanning-without.html, (08/01/07), November 2006.

[13] Jeremiah Grossman. I know if you]re logged-in, anywhere. [online], http://jeremiahgrossman.blogspot.com/2006/12/ i-know-if-youre-logged-in-anywhere.html, (08/08/07), December 2006.

[14] Jeremiah Grossman. I know where you]ve been. [online], http://jeremiahgrossman.blogspot.com/2006/08/i-know-where-youve-been.html, August 2006.

[15] Jeremiah Grossman, Robert Hansen, Petko Petkov, and Anton Rager. Cross Site Scripting Attacks: Xss Exploits and Defense. Syngress, 2007.

[16] Jeremiah Grossman and TC Niedzialkowski. Hacking intranet websites from the outside. Talk at Black Hat USA 2006, http://www.blackhat.com/presentations/bh-usa-06/ BH-US-06-Grossman.pdf, August 2006.

[17] Robert Hansen. Detecting firefox extentions. [online], http://ha.ckers.org/blog/20060823/detecting-firefox-extentions/, (08/08/07), August 2006.

[18] Philippe Le Hegaret, Ray Whitmer, and Lauren Wood. Document object model (dom). W3C recom- mendation, http://www.w3.org/DOM/, January 2005.

The bibliography

[20] Collin Jackson, Adam Barth, Andrew Bortz, Weidong Shao, and Dan Boneh. Protecting browsers from dns rebinding attack. In Proceedings of the 14th ACM Conference on Computer and Communication Security (CCS 07), October 2007.

[21] Markus Jakobsson and Sid Stamm. Invasive browser sniffing and countermeasures. In Proceedings of The 15th annual World Wide Web Conference (WWW2006), 2006.

[22] Martin Johns. (somewhat) breaking the same-origin policy by undermining dns-pinning. Posting to the Bugtraq mailinglist, http://www.securityfocus.com/archive/107/443429/30/180/threaded, August 2006.

[23] Martin Johns and Kanatoko Anvil. Using java in anti dns-pinning attacks (firefox and opera). [online], http://shampoo.antville.org/stories/1566124/, (08/27/07), Februar 2007.

[24] Martin Johns and Justus Winter. Requestrodeo: Client side protection against session riding. In Frank Piessens, editor, Proceedings of the OWASP Europe 2006 Conference, refereed papers track, Report CW448, pages 5 17. Departement Computerwetenschappen, Katholieke Universiteit Leuven, May 2006.

[25] Martin Johns and Justus Winter. Protecting the intranet against javascript malware and related at- tacks. In Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA 2007), July 2007.

[26] Dan Kaminsky. Black ops 2007: Design reviewing the web. talk at the Black Hat 2007 conference, http://www.doxpara.com/?q=node/1149, August 2007.

[27] Kanatoko. Anti-dns pinning + socket in flash. [online], http://www.jumperz.net/index.php?i=2&a=3&b=3, (19/01/07), January 2007.

The bibliography

[28] Chris Karlof, Umesh Shankar, J.D. Tygar, and David Wagner. Dynamic pharming attacks and the locked same-origin policies for web browsers. In Proceedings of the 14th ACM Conference on Com- puter and Communication Security (CCS]07), October 2007.

[29] Lars Kindermann. My address java applet. [online], http://reglos.de/myaddress/MyAddress.html (11/08/06), 2003.

[30] Amit Klein. Forging http request headers with flash actionscript. Whitepaper, http://www.securiteam.com/securityreviews/5KP0M1FJ5E.html, July 2006.

[31] SPI Labs. Detecting, analyzing, and exploiting intranet applications using javascript. Whitepaper, http://www.spidynamics.com/assets/documents/JSportscan.pdf, July 2006.

[32] SPI Labs. Stealing search engine queries with javascript. Whitepaper, http://www.spidynamics.com/assets/documents/JS SearchQueryTheft.pdf, 2006.

[33] V. T. Lam, Spyros Antonatos, P. Akritidis, and Kostas G. Anagnostakis. Puppetnets: Misusing web browsers as a distributed attack infrastructure. In Proceedings of the 13th ACM Conference on Com- puter and Communication Security (CCS]06), pages 221ȣ234, 2006.

[34] Julien Lamarre. Ajax without xmlhttprequest, frame, iframe, java or flash. [online], http://zingzoom.com/ajax/ajax with image.php, (02/02/2006), September 2005.

[35] Nathan McFeters and Billy Rios. Uri use and abuse. Whitepaper, http://www.xs-sniper.com/nmcfeters/URI Use and Abuse.pdf, July 2007.

[36] Haroon Meer and Marco Slaviero. It]s all about the timing... Whitepaper, http://www.sensepost.com/research/squeeza/dc-15-meer and slaviero-WP.pdf, Au- gust 2007.

The bibliography

[37] Adam Megacz. Firewall circumvention possible with all browsers. Posting to the Bugtraq mailing list, http://seclists.org/bugtraq/2002/Jul/0362.html, July 2002.

[38] Steffen Meschkat. Json rpc - cross site scripting and client side web services. Talk at the 23C3 Congress, http://events.ccc.de/congress/2006/Fahrplan/attachments/ 1198-jsonrpcmesch.pdf, December 2006.

[39] Samy. Technical explanation of the myspace worm. [online], http://namb.la/popular/tech.html, (01/10/06), October 2005.

[40] Mozilla Project. Mozilla port blocking. [online], http://www.mozilla.org/projects/ netlib/PortBanning.html (11/13/06), 2001.

[41] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear. Address allocation for private internets. RFC 1918, http://www.ietf.org/rfc/rfc1918.txt, February 1996.

[42] Jesse Ruderman. The same origin policy. [online], http://www.mozilla.org/projects/ security/components/same-origin.html (01/10/06), August 2001. 19

[43] Thomas Schreiber. Session riding - a widespread vulnerability in today¦s web applications. Whitepaper, SecureNet GmbH, http://www.securenet.de/papers/Session Riding.pdf, De- cember 2004.

[44] Princeton University Secure Internet Programming Group. Dns attack scenario. [online], http: //www.cs.princeton.edu/sip/news/dns-scenario.html, February 1996.

[45] Rajesh Sethumadhavan. Microsoft internet explorer local file accesses vulnerability. Posting to the full disclosure mailing list, http://seclists.org/fulldisclosure/2007/Feb/0434. html, February 2007.

The bibliography

[46] Josh Soref. Dns: Spoofing and pinning. [online], http://viper.haque.net/~timeless/blog/11/, (14/11/06), September 2003.

[47] Sid Stamm, Zulfikar Ramzan, and Markus Jakobsson. Drive-by pharming. Technical Report 641, Indiana University Computer Science, December 2006.

[48] Jochen Topf. The html form protocol attack. Whitepaper, http://www.remote.org/jochen/ sec/hfpa/hfpa.pdf, August 2001.

[49] Sergey Vzloman and Robert Hansen. Enumerate windows users in js. [online], http://ha.ckers.org/blog/20070518/enumerate-windows-users-in-js/, (08/08/07), May 2007.

[50] Sergey Vzloman and Robert Hansen. Read firefox settings (poc). [online], http://ha.ckers.org/blog/20070516/read-firefox-settings-poc/, (08/08/07), May 2007.