Examination of the Bitsquatting Attack Surface

Jaeson Schultz
Threat Research Engineer

April 2013

Agenda

- What is bitsquatting? Why should we care?
- Discussion of new bitsquatting attacks
- Bitsquatting statistics
- Mitigations for bitsquatting attacks

What is Bitsquatting?

Introduction

- Bitsquatting is a form of cybersquatting which specifically targets bit errors in computer memory
- A memory error occurs any time one or more bits being read from memory have changed state from what was previously written
- By changing a single bit, a target domain such as “twitter.com” can become the bitsquat domain “twitte2.com”
- An attacker can simply register a bitsquat domain, wait for a memory error to occur, and afterwards intercept traffic, infect the client, …

Binary	Oct	Dec	Hex	Glyph	Binary	Oct	Dec	Hex	Glyph	Binary	Oct	Dec	Hex	Glyph
010 0000	040	32	20	␣	100 0000	100	64	40	@	110 0000	140	96	60	`
010 0001	041	33	21	!	100 0001	101	65	41	A	110 0001	141	97	61	a
010 0010	042	34	22	"	100 0010	102	66	42	B	110 0010	142	98	62	b
010 0011	043	35	23	#	100 0011	103	67	43	C	110 0011	143	99	63	c
010 0100	044	36	24	$	100 0100	104	68	44	D	110 0100	144	100	64	d
010 0101	045	37	25	%	100 0101	105	69	45	E	110 0101	145	101	65	e
010 0110	046	38	26	&	100 0110	106	70	46	F	110 0110	146	102	66	f
010 0111	047	39	27	'	100 0111	107	71	47	G	110 0111	147	103	67	g
010 1000	050	40	28	(	100 1000	110	72	48	H	110 1000	150	104	68	h
010 1001	051	41	29	)	100 1001	111	73	49	I	110 1001	151	105	69	i
010 1010	052	42	2A	*	100 1010	112	74	4A	J	110 1010	152	106	6A	j
010 1011	053	43	2B	+	100 1011	113	75	4B	K	110 1011	153	107	6B	k
010 1100	054	44	2C	,	100 1100	114	76	4C	L	110 1100	154	108	6C	l
010 1101	055	45	2D	-	100 1101	115	77	4D	M	110 1101	155	109	6D	m
010 1110	056	46	2E	.	100 1110	116	78	4E	N	110 1110	156	110	6E	n
010 1111	057	47	2F	/	100 1111	117	79	4F	O	110 1111	157	111	6F	o
011 0000	060	48	30	0	101 0000	120	80	50	P	111 0000	160	112	70	p
011 0001	061	49	31	1	101 0001	121	81	51	Q	111 0001	161	113	71	q
011 0010	062	50	32	2	101 0010	122	82	52	R	111 0010	162	114	72	r
011 0011	063	51	33	3	101 0011	123	83	53	S	111 0011	163	115	73	s
011 0100	064	52	34	4	101 0100	124	84	54	T	111 0100	164	116	74	t
011 0101	065	53	35	5	101 0101	125	85	55	U	111 0101	165	117	75	u
011 0110	066	54	36	6	101 0110	126	86	56	V	111 0110	166	118	76	v
011 0111	067	55	37	7	101 0111	127	87	57	W	111 0111	167	119	77	w
011 1000	070	56	38	8	101 1000	130	88	58	X	111 1000	170	120	78	x
011 1001	071	57	39	9	101 1001	131	89	59	Y	111 1001	171	121	79	y
011 1010	072	58	3A	:	101 1010	132	90	5A	Z	111 1010	172	122	7A	z

Causes of computer memory errors

- Cosmic Rays
  High energy particles that strike the Earth as frequently as 10,000 per square meter per second
- Heat
  Operating a device outside the recommended operating environment.
- Nuclear Explosions
  Intense neutron emission from low yield nuclear explosions induce a sharp increase in the frequency of bitsquat requests
- Defects in Manufacturing
  Errors in memory have been traced to alpha particle emissions from chip packaging materials.

Previous Research on Bitsquatting

- Original bitsquatting research was by Artem Dinaburg from Raytheon. He focused on bit flips in the characters of 2nd level domains and demonstrated that bitsquatting works!
- Python code was released by Dinaburg to compute bitsquats:

```
def is_valid(charnum):
    if  (charnum >= ord('0') and charnum <= ord('9')) or\
        (charnum >= ord('a') and charnum <= ord('z')) or\
        (charnum >= ord('A') and charnum <= ord('Z')) or\
        charnum == ord('-'):
        return True
    else:
        return False
```

- Duane Wessels of VeriSign followed up with bitsquatting research that confirmed that the bit errors were not a result of the network DNS resolution process. (UDP checksums work)

New Bitsquatting Attacks

Subdomain Delimiter Bitsquats

- RFC1035 declared the valid syntax for domain name labels, which was later refined under RFC1123. According to these RFCs, the only valid characters inside a domain name are:
  1. A-Z
  2. a-z
  3. 0-9
  4. - (hyphen)
- The dots “.” which separate subdomains can also experience bit errors that cause them to become a lowercase ‘n’ and vice versa.

010 1110	056	46	2E	.	110 1110	156	110	6E	n

Subdomain Delimiters: “n” flips to “.”

- If a second level domain name contains the letter “n” and there are two or more characters after the letter “n”, then this is a potential bitsquat.
- Two example domains:
  “windowsupdate.com” has bitsquat “dowsupdate.com”
  “symantecliveupdate.com” has bitsquat “tecliveupdate.com”

2/26/13 5:21:25.000 PM
```
client 68.87.68.174#52076: query: download.wi.dowsupdate.com IN A -ED (198.23.252.184)
client 68.87.68.174#17467: query: download.wi.dowsupdate.com IN A -ED (198.23.252.184)
client 68.87.68.174#16820: query: download.wi.dowsupdate.com IN A -ED (198.23.252.184)
client 68.87.68.174#58590: query: download.wi.dowsupdate.com IN A -ED (198.23.252.184)
client 76.96.90.215#43579: query: download.wi.dowsupdate.com IN A -ED (198.23.252.184)
client 76.96.90.215#55497: query: download.wi.dowsupdate.com IN A -ED (198.23.252.184)
client 76.96.90.215#41264: query: download.wi.dowsupdate.com IN A -ED (198.23.252.184)
client 76.96.90.215#55944: query: download.wi.dowsupdate.com IN A -ED (198.23.252.184)
client 76.96.90.215#37722: query: download.wi.dowsupdate.com IN A -ED (198.23.252.184)
client 76.96.90.215#62119: query: download.wi.dowsupdate.com IN A -ED (198.23.252.184)
```
Show all 36 lines
host=data.0xfeedcafe.com ▾ | sourcetype=query.log ▾ | source=/var/log/query.log ▾

2/26/13 5:21:24.000 PM
```
client 68.87.68.174#32447: query: download.wi.dowsupdate.com IN A -ED (198.23.252.184)
client 68.87.68.174#56039: query: download.wi.dowsupdate.com IN A -ED (198.23.252.184)
client 68.87.68.174#61187: query: download.wi.dowsupdate.com IN A -ED (198.23.252.184)
client 68.87.68.174#53353: query: download.wi.dowsupdate.com IN A -ED (198.23.252.184)
```
host=data.0xfeedcafe.com ▾ | sourcetype=query.log ▾ | source=/var/log/query.log ▾

2/26/13 5:11:32.000 PM
```
client 77.88.44.250#5335: query: ns2.dowsupdate.com IN A -ED (198.23.252.184)
client 77.88.44.250#5335: query: ns2.dowsupdate.com IN A -ED (198.23.252.184)
```
host=data.0xfeedcafe.com ▾ | sourcetype=query.log ▾ | source=/var/log/query.log ▾

2/26/13 5:11:32.000 PM
```
client 213.180.209.250#5335: query: ns2.dowsupdate.com IN A -ED (198.23.252.184)
client 213.180.209.250#5335: query: ns2.dowsupdate.com IN A -ED (198.23.252.184)
client 77.88.43.250#5335: query: ns2.dowsupdate.com IN A -ED (198.23.252.184)
client 77.88.43.250#5335: query: ns2.dowsupdate.com IN A -ED (198.23.252.184)
```
host=data.0xfeedcafe.com ▾ | sourcetype=query.log ▾ | source=/var/log/query.log ▾

2/26/13 5:01:23.000 PM
```
client 76.96.90.217#61851: query: download.wi.dowsupdate.com IN A -ED (198.23.252.184)
client 76.96.90.217#44091: query: download.wi.dowsupdate.com IN A -ED (198.23.252.184)
client 76.96.90.217#64407: query: download.wi.dowsupdate.com IN A -ED (198.23.252.184)
client 76.96.90.217#45463: query: download.wi.dowsupdate.com IN A -ED (198.23.252.184)
client 68.87.68.165#29197: query: download.wi.dowsupdate.com IN A -ED (198.23.252.184)
client 68.87.68.165#61771: query: download.wi.dowsupdate.com IN A -ED (198.23.252.184)
client 68.87.68.165#50891: query: download.wi.dowsupdate.com IN A -ED (198.23.252.184)
client 68.87.68.165#30059: query: download.wi.dowsupdate.com IN A -ED (198.23.252.184)
client 68.87.68.165#32198: query: download.wi.dowsupdate.com IN A -ED (198.23.252.184)
client 68.87.68.165#28906: query: download.wi.dowsupdate.com IN A -ED (198.23.252.184)
```
Show all 28 lines
host=data.0xfeedcafe.com ▾ | sourcetype=query.log ▾ | source=/var/log/query.log ▾

3/9/13 3:45:40.000 AM	client 124.238.215.152#48775: query: ns2.tecliveupdate.com IN AAAA -EDC (198.23.252.184) client 124.238.215.152#41522: query: ns1.tecliveupdate.com IN AAAA -EDC (198.23.252.184) client 124.238.215.152#49027: query: liveupdatg.syma.tecliveupdate.com IN A -EDC (198.23.252.184) host=data.0xfeedcafe.com ▾ | sourcetype=query.log ▾ | source=/var/log/query.log ▾ | dns_client_ip=124.238.215.152 ▾
2/26/13 10:42:50.000 PM	client 23.67.252.244#32818: query: liveupdatg.syma.tecliveupdate.com IN A -ED (198.23.252.184) host=data.0xfeedcafe.com ▾ | sourcetype=query.log ▾ | source=/var/log/query.log ▾ | dns_client_ip=23.67.252.244 ▾
1/21/13 5:43:04.000 AM	client 46.229.154.12#51678: query: liveupdate.syma.tecliveupdate.com IN A -ED (198.23.252.184) host=data.0xfeedcafe.com ▾ | sourcetype=query.log ▾ | source=/var/log/query.log ▾ | dns_client_ip=46.229.154.12 ▾
1/7/13 5:55:41.000 PM	client 24.143.205.45#45698: query: liveupdate.syma.tecliveupdate.com IN A -ED (198.23.252.184) host=data.0xfeedcafe.com ▾ | sourcetype=query.log ▾ | source=/var/log/query.log ▾ | dns_client_ip=24.143.205.45 ▾
1/2/13 8:14:46.000 PM	client 204.0.54.102#56624: query: `liveupdate.syma.tecliveupdate.com IN A -ED (198.23.252.184) host=data.0xfeedcafe.com ▾ | sourcetype=query.log ▾ | source=/var/log/query.log ▾ | dns_client_ip=204.0.54.102 ▾

Subdomain Delimiters: “.” flips to “n”

- If a 2nd level domain name uses 3rd level subdomains, these can be leveraged into bitsquat domains by replacing the “.” separating the 3rd and 2nd level domain labels with the letter “n”.
- Three example domains:

  “s.ytimg.com” has bitsquat “snytimg.com”
  “mail.google.com” has bitsquat “mailngoogle.com”
  “state.ny.us” has bitsquat “statenny.us”


3/12/13 6:47:18.000 AM
[Tue Mar 12 11:47:18 2013] [error] [client 77.28.78.34] File does not exist: /var/www/yts, referer:
http://www.youtube.com/results?search_query=will.i.am+ft+britney+spears+scream+and+shout&oq=Will.I.A
reduced.1.0.0l4.123187.123187.0.125978.1.1.0.0.0.0.284.284.2-1.1.0...0.0...1ac.1.h79oQXeA9s4
host=data.0xfeedcafe.com | sourcetype=apache_error | source=/var/log/apache2/error.log
3/12/13 6:47:18.000 AM
77.28.78.34 - - [12/Mar/2013:11:47:18 +0000] "GET /yts/img/pixel-vfl3z5WfW.gif HTTP/1.1" 404 516
"http://www.youtube.com/results?search_query=will.i.am+ft+britney+spears+scream+and+shout&oq=Will.I.
reduced.1.0.0l4.123187.123187.0.125978.1.1.0.0.0.0.284.284.2-1.1.0...0.0...1ac.1.h79oQXeA9s4" "Mozil
AppleWebKit/537.1 (KHTML, like Gecko) Chrome/21.0.1180.89 Safari/537.1" "snytimg.com"
host=data.0xfeedcafe.com | sourcetype=access_combined | source=/var/log/apache2/access.log
3/12/13 6:47:18.000 AM
client 62.162.32.10#32839: query: snytimg.com IN A -ED (198.23.252.184)
host=data.0xfeedcafe.com | sourcetype=query.log | source=/var/log/query.log | dns_client_ip=62.162.32.10

3/31/13 9:18:16.000 PM
184.153.66.222 - - [01/Apr/2013:02:18:16 +0000] "GET /favicon.ico HTTP/1.1" 200 1445 "http://mailngoogle.com/" "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_6_3; en-us; Silk/1.0.22.153_10033210) AppleWebKit/533.16 (KHTML, like Gecko) Version/5.0 Safari/533.16 Silk-Accelerated=true" "mailngoogle.com"
host=data.0xfeedcafe.com | sourcetype=access_combined | source=/var/log/apache2/access.log

3/31/13 9:18:16.000 PM
184.153.66.222 - - [01/Apr/2013:02:18:16 +0000] "GET /feedcafe.logo.png HTTP/1.1" 200 7413 "http://mailngoogle.com/" "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_6_3; en-us; Silk/1.0.22.153_10033210) AppleWebKit/533.16 (KHTML, like Gecko) Version/5.0 Safari/533.16 Silk-Accelerated=true" "mailngoogle.com"
host=data.0xfeedcafe.com | sourcetype=access_combined | source=/var/log/apache2/access.log

3/31/13 9:18:16.000 PM
184.153.66.222 - - [01/Apr/2013:02:18:16 +0000] "GET /binary-bkg.png HTTP/1.1" 200 988 "http://mailngoogle.com/" "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_6_3; en-us; Silk/1.0.22.153_10033210) AppleWebKit/533.16 (KHTML, like Gecko) Version/5.0 Safari/533.16 Silk-Accelerated=true" "mailngoogle.com"
host=data.0xfeedcafe.com | sourcetype=access_combined | source=/var/log/apache2/access.log

3/31/13 9:18:16.000 PM
184.153.66.222 - - [01/Apr/2013:02:18:16 +0000] "GET / HTTP/1.1" 200 750 "-" "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_6_3; en-us; Silk/1.0.22.153_10033210) AppleWebKit/533.16 (KHTML, like Gecko) Version/5.0 Safari/533.16 Silk-Accelerated=true" "mailngoogle.com"
host=data.0xfeedcafe.com | sourcetype=access_combined | source=/var/log/apache2/access.log

3/31/13 9:18:16.000 PM
client 24.92.226.208#2869: query: mailngoogle.com IN A -ED (198.23.252.184)
host=data.0xfeedcafe.com | sourcetype=query.log | source=/var/log/query.log | dns_client_ip=24.92.226.208

cisco

3/8/13 8:40:30.000 AM
client 74.125.189.22#56927: query: omh.statenny.us IN MX - (198.23.252.184)
host=data.0xfeedcafe.com ▾ | sourcetype=query.log ▾ | source=/var/log/query.log ▾ | dns_client_ip=74.125.189.22 ▾

3/8/13 8:40:30.000 AM
client 62.183.62.117#20686: query: NS2.statenny.us IN AAAA -EDC (198.23.252.184)
client 62.183.62.117#24288: query: NS1.statenny.us IN AAAA -EDC (198.23.252.184)
client 62.183.62.117#54780: query: omh.statenny.us IN MX -EDC (198.23.252.184)
client 74.125.18.215#48648: query: omh.statenny.us IN MX - (198.23.252.184)
host=data.0xfeedcafe.com ▾ | sourcetype=query.log ▾ | source=/var/log/query.log ▾ | dns_client_ip=62.183.62.117 ▾

cisco

URL Delimiter Bitsquats

- One of the most popular contexts for a domain name to appear is inside of a URL, especially over HTTP. For example, look at the popularity of the bitsquat domains which was originally published by Dinaburg in his 2011 research paper:

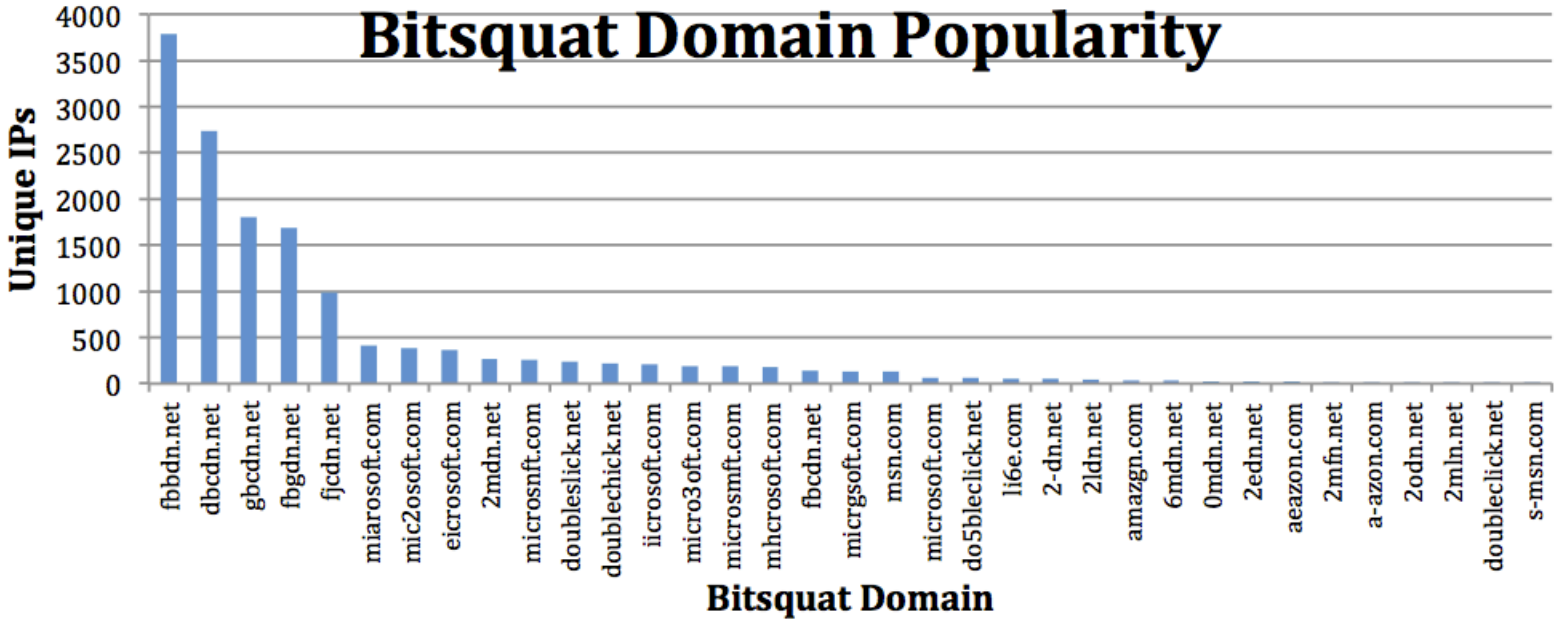


Figure 9: Domain popularity, ordered by the number of unique IPs with the domain in the HTTP `Host` header.

URL Delimiter Bitsquats

- Inside a URL or href, forward slash characters “/” will act as delimiters separating the scheme from the hostname from the URL path
- The letter “o” can flip a bit to become the forward slash character “/”, cutting short the domain name.

010 1111	057	47	2F	/	110 1111	157	111	6F	o

URL Delimiters: “o” flips to “/”

- Possible when a domain name in a URL contains the letter “o” and the preceding characters form a valid second level domain name.
- Most interesting aspect of this method is that domains at non-public Top Level Domains (TLDs) can be targeted.
- A few examples:
  “tcoss.scott.af.mil” has bitsquat “tcoss.sc”
  “bop.peostri.army.mil” has bitsquat “bop.pe”
  “ecampus.phoenix.edu” has bitsquat “ecampus.ph”
  “trading.scottrade.com” has bitsquat “trading.sc”

3/2/13 2:39:18.000 PM
75.46.29.137 - - [02/Mar/2013:19:39:18 +0000] "GET / HTTP/1.1" 200 750 "-" "Mozilla/5.0 (Linux; Android 4.0.4; ZTE N9120 Build/IMM76I) AppleWebKit/535.19 (KHTML, like Gecko) Chrome/18.0.1025.166 Mobile Safari/535.19" "ecampus.ph"
host=data.0xfeedcafe.com | sourcetype=access_combined | source=/var/log/apache2/access.log

3/2/13 2:39:17.000 PM
client 206.141.193.32#37340: query: ecampus.ph IN A -EDC (198.23.252.184)
host=data.0xfeedcafe.com | sourcetype=query.log | source=/var/log/query.log

3/4/13 8:48:11.000 PM
76.18.128.127 - - [05/Mar/2013:01:48:11 +0000] "GET / HTTP/1.1" 200 750 "-" "Mozilla/5.0 (Linux; Android 4.2.2; Nexus 7 Build/JDQ39) AppleWebKit/535.19 (KHTML, like Gecko) Chrome/18.0.1025.166 Safari/535.19" "trading.sc"
host=data.0xfeedcafe.com | sourcetype=access_combined | source=/var/log/apache2/access.log

3/4/13 8:48:10.000 PM
client 76.96.90.216#43923: query: trading.sc IN A -ED (198.23.252.184)
host=data.0xfeedcafe.com | sourcetype=query.log | source=/var/log/query.log | dns_client_ip=76.96.90.216

cisco

URL Delimiters: “/” flips to “o”

- Sometimes the slashes inside a URL or href can experience a bit error and become a lowercase letter “o”
- URLs will contain at least 2 slashes
- First 2 slashes separate scheme from hostname, a third slash may separate the hostname from the path.

  http://www.cisco.com/
- Generally only bit flips of the second slash produce viable bitsquat domains

URL Delimiters: Bad syntax is OK

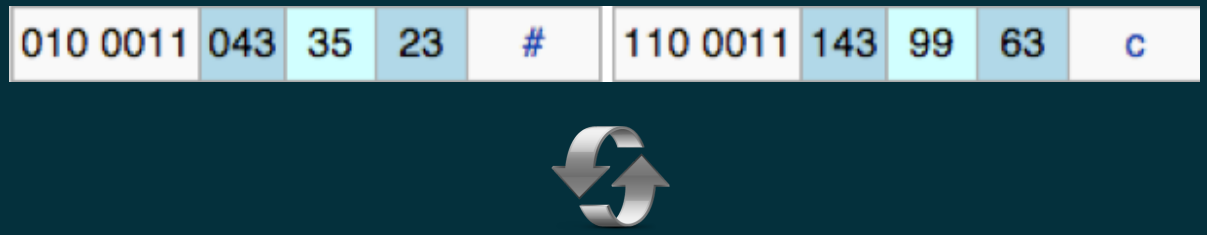
- Some domains do not use 3rd level subdomain names. For example, the domain slashdot.org. When a domain does not use 3rd level subdomains, it can be vulnerable to some additional types of URL delimiter bitsquats. For example consider the URL: http://slashdot.org/
- After a bit error in the second forward slash the URL becomes: http:/oslashdot.org/
- Though the syntax is not valid, your web browser will typically helpfully correct for the missing slash and direct traffic to the domain “oslashdot.org”.
- As of December 2012 the team from no-www.org have catalogued 60,000 domains that do not use 3rd level subdomains

URL Delimiters: Bad syntax is OK

- If a domain name does not use 3rd level subdomains and begins with the letter “o” it can be susceptible to another obscure form of URL delimiter bitsquat. For example consider the URL: http://oreilly.com/
- After a bit error the URL becomes: http:///reilly.com/
- Though the syntax is not valid, your web browser will helpfully correct for the extra slash and direct traffic to the domain “reilly.com”

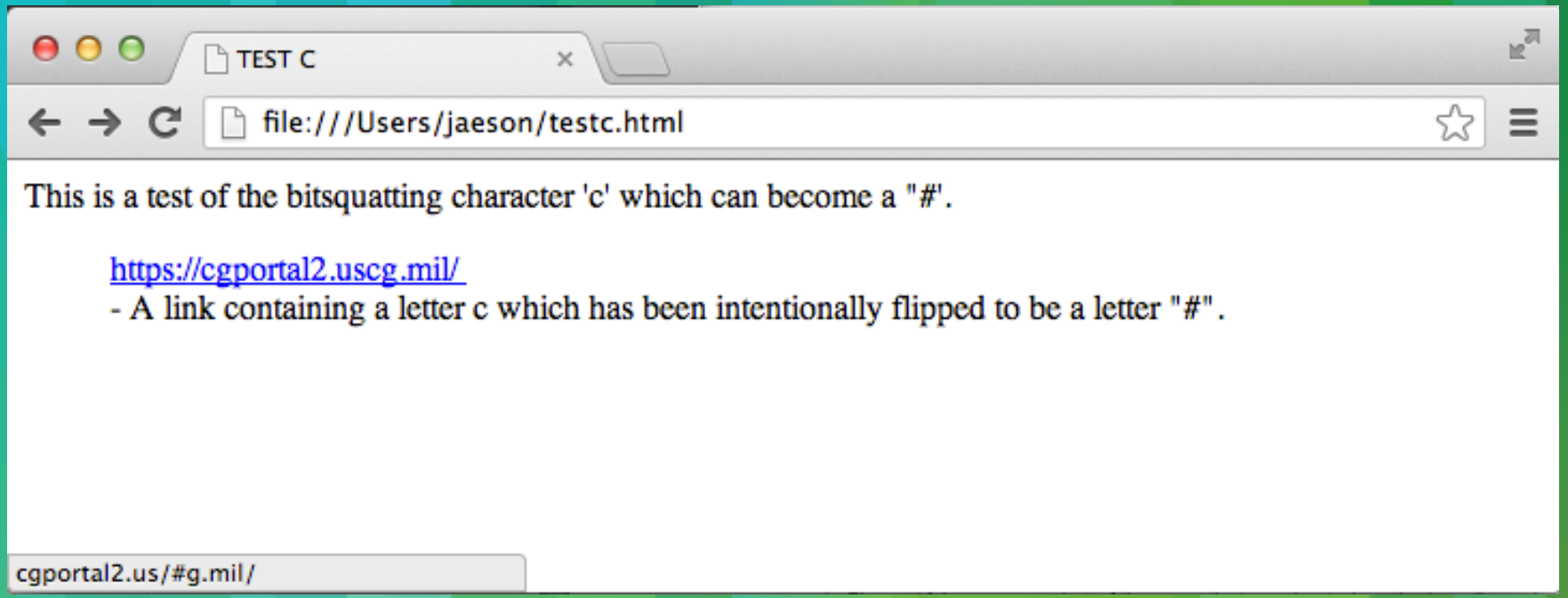
More URL Delimiter Bitsquats

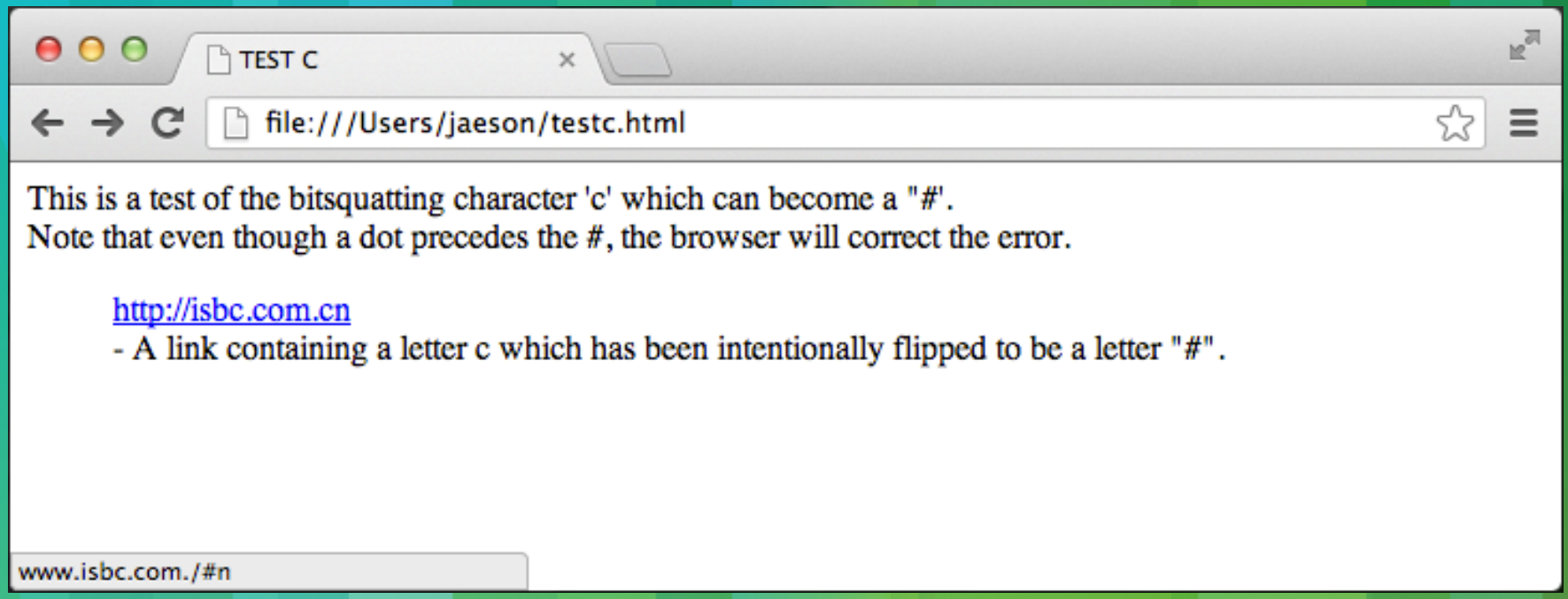
- Inside a URL, anchor characters “#” will act as delimiters separating the the hostname from the anchor location on the web page
- The letter “c” can flip a bit to become the pound character “#”, cutting short the domain name.



URL Delimiters: “c” flips to “#”

- Attack is possible when a domain in a URL contains the letter “c” and the preceding characters form a valid second level domain name. Bad syntax is, like before, OK.
- Similar to the “o” ➔ ”/” bitsquats, this method allows for targeting domains at non-public Top Level Domains (TLDs) such as .mil, .gov, .edu.
- Some examples:
  “cgportal2.uscg.mil” has bitsquat “cgportal2.us”
  “certauth.bechtel.com” has bitsquat “certauth.be”
  “emergency.cdc.gov” has bitsquat “emergency.cd”
  “pki.nrc.gov” has bitsquat “pki.nr”
  “isbc.com.cn” has bitsquat “isbc.com”


TEST C
file:///Users/jaeson/testc.html
This is a test of the bitsquatting character 'c' which can become a "#'.
https://cgportal2.uscg.mil/
- A link containing a letter c which has been intentionally flipped to be a letter "#".
cgportal2.us/#g.mil/


TEST C
file:///Users/jaeson/testc.html
This is a test of the bitsquatting character 'c' which can become a "#".
Note that even though a dot precedes the #, the browser will correct the error.
http://isbc.com.cn
- A link containing a letter c which has been intentionally flipped to be a letter "#".
www.isbc.com./#n

Top Level Domain (TLD) Bitsquats

- Bit errors can occur anywhere, so why not in the TLD?
- Most of the generic TLDs (gTLDs) have no bitsquats whatsoever.
- Two gTLDs contain URL Delimiter type bitsquats stemming from the presence of the letter “o”. These are the gTLDs “.pro” and “.coop” with corresponding URL delimiter type bitsquats at the country code TLDs (ccTLDs): .pr (Puerto Rico) and .co (Colombia) respectively
- It’s a positive thing then that relatively few URLs exist that point to .pro and .coop

Top Level Domain (ccTLD) Bitsquats

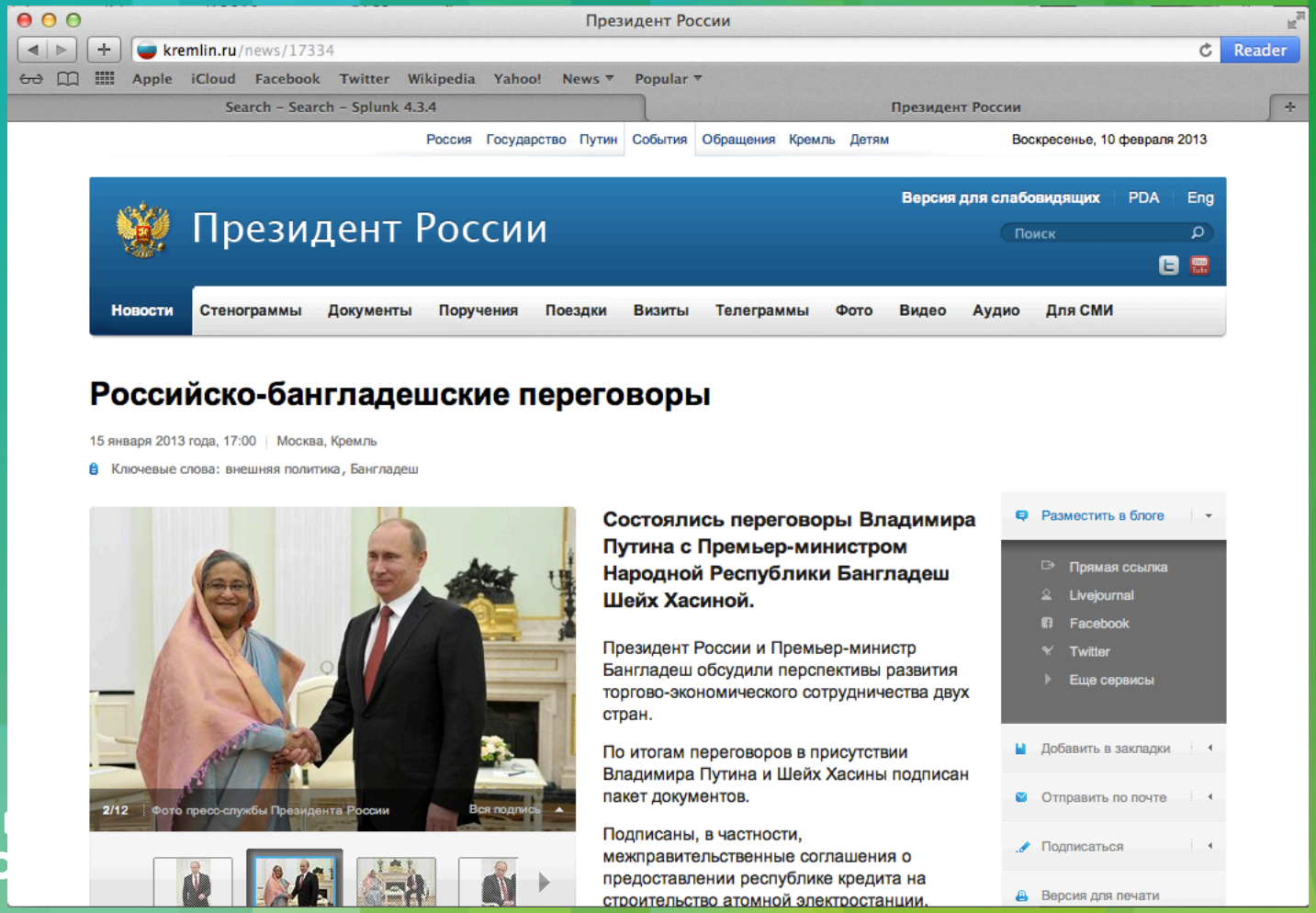
- There happen to be several ccTLDs where bitsquats exist. It is interesting to note that some ccTLDs have no valid bitsquats while other ccTLDs have many. After surveying all valid Internet TLDs and checking the number of possible bitsquats, the following was found:

  - All 44 Internationalized Domain Name (IDN) TLDs have zero bitsquats
  - 4 ccTLDs have zero bitsquats (nl – Netherlands, py – Paraguay, uy – Uruguay, za – South Africa)
  - 15 ccTLDs have one bitsquat (incl. uk – United Kingdom, hk – Hong Kong)
  - 33 ccTLDs have two bitsquats (incl. us – United States, de – Germany, jp – Japan)
  - 43 ccTLD have three bitsquats (incl. fr – France, no – Norway, va – Vatican
  - 56 ccTLDs have four bitsquats (incl. ru – Russia, kr – South Korea)
  - 43 ccTLDs have five bitsquats (incl. ca – Canada, it – Italy, eu – Europe)
  - 37 ccTLDs have six bitsquats (incl. es – Spain, gr – Greece, in – India)
  - 14 ccTLDs have seven bitsquats (incl. co – Colombia, ch – Switzerland)
  - 2 ccTLDs have eight bitsquats (cm – Cameroon, cn – China)
  - 1 ccTLD has nine bitsquats (cg – Republic of Congo)
  - 1 ccTLD has ten bitsquats (ci – Ivory Coast)

1/28/13 2:01:39.000 AM
180.234.143.197 - - [28/Jan/2013:07:01:39 +0000] "GET /news/17334 HTTP/1.1" 404 501 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.6; rv:2.0b6pre) Gecko/20100908 Firefox/4.0b6pre" "kremlin.re"
host=data.0xfeedcafe.com | sourcetype=access_combined | source=/var/log/apache2/access.log

1/28/13 2:01:39.000 AM
[Mon Jan 28 07:01:39 2013] [error] [client 180.234.143.197] File does not exist: /var/www/news
host=data.0xfeedcafe.com | sourcetype=apache_error | source=/var/log/apache2/error.log

1/28/13 2:01:38.000 AM
client 180.234.0.193#22285: query: kremlin.re IN AAAA -EDC (198.23.252.184)
client 180.234.0.197#24213: query: kremlin.re IN A -EDC (198.23.252.184)
host=data.0xfeedcafe.com | sourcetype=query.log | source=/var/log/query.log | dns_client_ip=180.234.0.193



2/26/13 8:38:18.000 AM	client 173.194.96.16#37953: query: cpvo.europa.mu IN MX - (198.23.252.184)	host=data.0xfeedcafe.com ▾ | sourcetype=query.log ▾ | source=/var/log/query.log ▾
2/26/13 6:07:56.000 AM	client 202.123.2.17#52937: query: ec.europa.mu IN MX -ED (198.23.252.184)	host=data.0xfeedcafe.com ▾ | sourcetype=query.log ▾ | source=/var/log/query.log ▾
2/26/13 12:59:42.000 AM	client 8.0.18.115#63518: query: ec.europa.mu IN MX -EDC (198.23.252.184)	host=data.0xfeedcafe.com ▾ | sourcetype=query.log ▾ | source=/var/log/query.log ▾
2/25/13 8:04:59.000 PM	client 208.67.217.21#38522: query: cpvo.europa.mu IN MX - (198.23.252.233)	host=data.0xfeedcafe.com ▾ | sourcetype=query.log ▾ | source=/var/log/query.log ▾
2/25/13 4:09:16.000 PM	client 74.125.189.21#36803: query: ec.europa.mu IN MX - (198.23.252.233)	host=data.0xfeedcafe.com ▾ | sourcetype=query.log ▾ | source=/var/log/query.log ▾
2/24/13 7:32:13.000 PM	client 8.0.16.26#39871: query: ext.eeas.europa.mu IN MX -EDC (198.23.252.184)	host=data.0xfeedcafe.com ▾ | sourcetype=query.log ▾ | source=/var/log/query.log ▾
2/24/13 6:07:20.000 PM	client 74.125.191.16#58256: query: eeas.europa.mu IN MX - (198.23.252.233)	host=data.0xfeedcafe.com ▾ | sourcetype=query.log ▾ | source=/var/log/query.log ▾
2/24/13 4:48:08.000 AM	client 208.69.35.21#42293: query: ext.eeas.europa.mu IN MX - (198.23.252.184)	host=data.0xfeedcafe.com ▾ | sourcetype=query.log ▾ | source=/var/log/query.log ▾

1/7/13 7:13:50.000 AM
client 77.87.224.149#1243: query: bk.bund.ee IN AAAA -EDC (198.23.252.184)
host=data.0xfeedcafe.com | sourcetype=query.log | source=/var/log/query.log | dns_client_ip=77.87.224.149

12/22/12 2:02:20.000 AM
client 77.87.228.44#62091: query: polizei.bund.ee IN AAAA -EDC (198.23.252.184)
client 77.87.224.149#5384: query: polizei.bund.ee IN A -EDC (198.23.252.184)
host=data.0xfeedcafe.com | sourcetype=query.log | source=/var/log/query.log | dns_client_ip=77.87.228.44

3/9/13 12:57:07.000 AM
client 8.0.18.41#24329: query: wsv.bund.dm IN MX -EDC (198.23.252.184)
host=data.0xfeedcafe.com | sourcetype=query.log | source=/var/log/query.log | dns_client_ip=8.0.18.41

3/9/13 12:50:31.000 AM
client 74.125.16.81#61100: query: wsv.bund.dm IN MX - (198.23.252.184)
host=data.0xfeedcafe.com | sourcetype=query.log | source=/var/log/query.log | dns_client_ip=74.125.16.81

3/8/13 3:43:15.000 PM
client 64.246.165.160#15786: query: WWW.BUND.DM IN A - (198.23.252.233)
host=data.0xfeedcafe.com | sourcetype=query.log | source=/var/log/query.log | dns_client_ip=64.246.165.160

New Generic TLD (gTLD) Bitsquats

- In 2013 ICANN is approving a number of new gTLDs. Some of these proposed new gTLDs contain subdomain delimiter bitsquats for the entire TLD. Possessing one of these would allow the attacker to mount a bitsquat attack against all domains registered under the target gTLD.

.cleaning -> clea.ing (new gTLD .ing)
.exchange -> excha.ge (Georgia)
.helsinki -> helsi.ki (Kiribati)
.holdings -> holdi.gs (S.Georgia and S.Sandwich Islands)
.international -> internatio.al (Albania)
.tennis -> ten.is (Iceland)

New Generic TLD (gTLD) Bitsquats

- Several of the proposed new gTLDs will have letter “o” based URL delimiter bitsquats in ccTLD space

.boo -> .bo (Bolivia)
.bio -> .bi (Burundi)
.cooking -> .co (Colombia)
.cool -> .co (Colombia)
.cloud -> .cl (Chile)
.ecom -> .ec (Ecuador)
.food -> .fo (Faroe Islands)
.football -> .fo (Faroe Islands)
.global -> .gl (Greenland)
.kyoto -> .ky (Cayman Islands)
.ngo -> .ng (Nigeria)
.photo -> .ph (Philippines)
.photography -> .ph (Philippines)
.photos -> .ph (Philippines)
.prof -> .pr (Puerto Rico)
.property -> .pr (Puerto Rico)
.properties -> .pr (Puerto Rico)
.scot -> .sc (Seychelles)
.shop -> .sh (St. Helena)

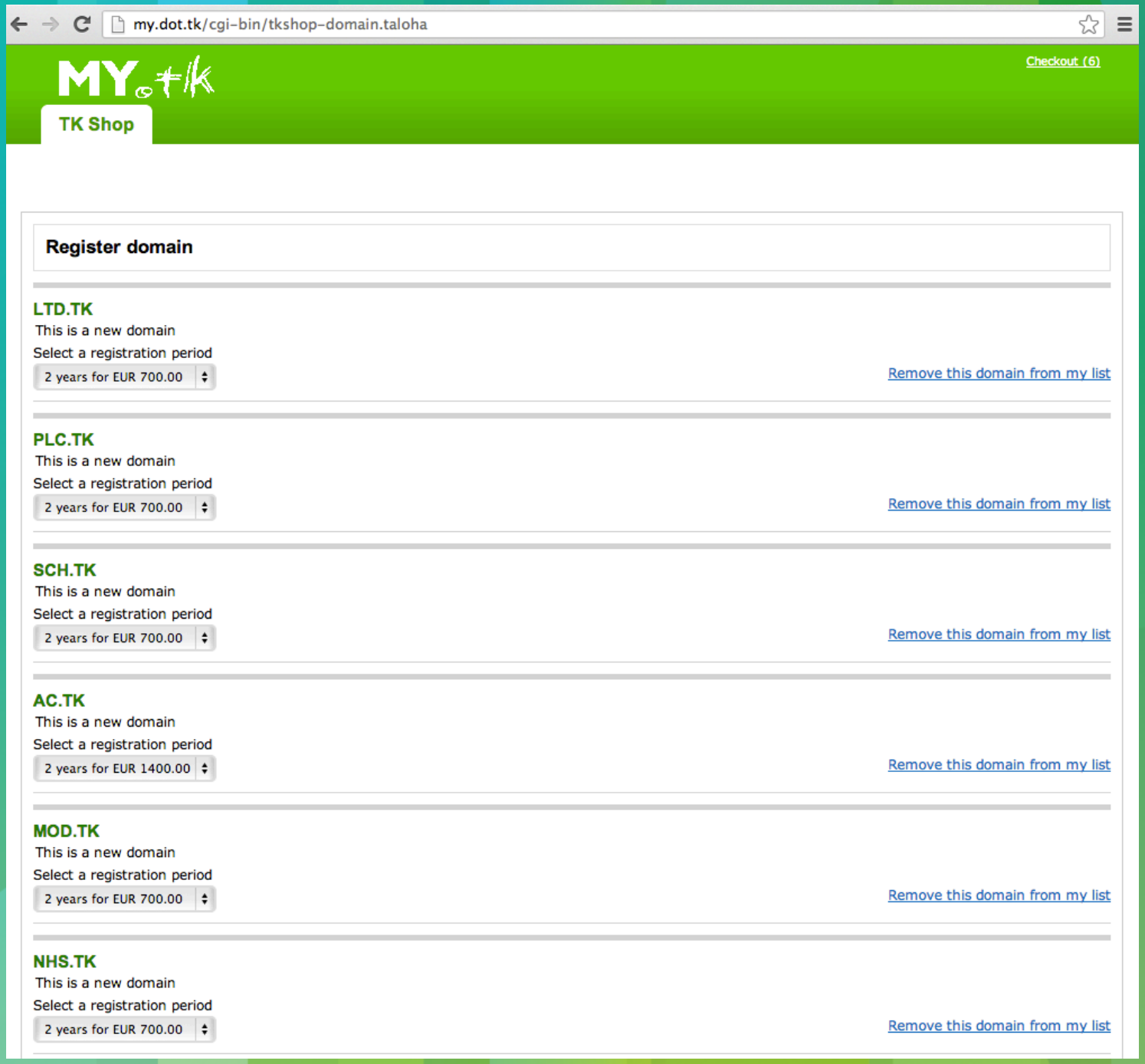
New Generic TLD (gTLD) Bitsquats

- Several of the proposed new gTLDs will also have letter “c” based URL delimiter bitsquats in ccTLD space

.rocks -> .ro (Romania)
.auction -> .au (Austrailia)
.doctor -> .do (Dominican Republic)
.accountant -> .ac
.archi -> .ar (Argentina)
.architect -> .ar (Argentina)
.recipes -> .re (Reunion Island)
.soccer -> .so (Somalia)
.inc -> .in (India)

More ccTLD Bitsquats

- The ".uk" (United Kingdom) ccTLD has one ccTLD bitsquat.
- The bitsquat ccTLD is ".tk" (Tokelau)
- The .uk domain registrar Nominet restricts .uk domain names to one of 13 2nd level domain prefixes. For example, co.uk, net.uk, org.uk, and so on.
- 6 of the 13 2nd level .tk domains are available. By registering one of these domains, a bitsquatter would receive bitsquats for any domain underneath the corresponding 2nd level domain in .uk


my.dot.tk/cgi-bin/tkshop-domain.taloha
MY.TK
TK Shop
Checkout (6)
Register domain
LTD.TK
This is a new domain
Select a registration period
2 years for EUR 700.00
Remove this domain from my list
PLC.TK
This is a new domain
Select a registration period
2 years for EUR 700.00
Remove this domain from my list
SCH.TK
This is a new domain
Select a registration period
2 years for EUR 700.00
Remove this domain from my list
AC.TK
This is a new domain
Select a registration period
2 years for EUR 1400.00
Remove this domain from my list
MOD.TK
This is a new domain
Select a registration period
2 years for EUR 700.00
Remove this domain from my list
NHS.TK
This is a new domain
Select a registration period
2 years for EUR 700.00
Remove this domain from my list

Bitsquatting Archaeology

- Looking at the domain whois records for some of the bitsquat domains that have already been registered yields some interesting findings:
  - The bitsquat domain wwwnfacebook.com was registered back in 2009, a full 2 years before the initial research paper on bitsquatting was published
  - The bitsquat domain otwitter.com was registered in 2009.
  - The bitsquat domain wwwnyoutube.com was registered in 2009.
- Some of the earliest bitsquat domain registrations, such as wwwnfacebook.com have come from "domainers" --organizations that register domain names to place ads or redirect traffic for profit.

  These domainers essentially noticed and capitalized on traffic destined for bitsquat domains likely unaware of the reason for the traffic.

Current Bitsquatting Mitigations

- Use Error Correcting (ECC) memory.
  Needs to happen simultaneously, and world-wide to be an effective solution.
- Register the bitsquat domain so that no third party can register it.
  This is not always possible, as many popular bitsquat domains have already been registered. Depending on the length of the domain, this can also be a costly endeavor.
- We can do better than this…

New Mitigation for Bitsquatting #1

- Because some of these new bitsquatting techniques rely on 3rd level domain names to work, then a careful strategy around their selection and use can help avoid the possibility of bitsquats
- Subdivide 2nd level domain traffic among a large number of 3rd level domains. Each subdomain takes on a small slice of the overall potential bitsquat traffic and therefore becomes much less likely to result in a successful bitsquat attack. Using a large number of subdomains creates much more work and expense for a potential bitsquat attacker.
- If additionally those subdomains are changed or updated with any frequency, a bitsquatter will have practically no chance at a successful attack.

- Amazon includes in their web pages content from a domain named cloudfront.com. The 3rd level domain names here normally would make great URL delimiter bitsquats because the “o” in cloudfront yields a valid ccTLD in .cl (Chile)

- Amazon changes the subdomain at cloudfront.com frequently enough that this thwarts attempts to capitalize on bitsquat traffic. By changing the 3rd level domain name frequently enough Amazon leaves too small a window of time in which to set-up and collect bitsquat traffic.

```
1283 var adcode;
1284 if ((0+6) <= getFlashVer()) {
1285   var flashVars = getFlashVarsStr();
1286   adcode = get3pPixed();
1287   adcode += '<OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000" ID=FLASH_AD WIDTH="300" HEIGHT="250"><PARAM
     NAME=movie VALUE="//d2o307dm5mqftz.cloudfront.net/1505855001/1357341372265/Shipping_C.swf"><param name="flashvars"
     value="'+ flashVars + '"><PARAM NAME=quality VALUE=high><PARAM NAME=bgcolor VALUE=#FFFFFF><PARAM NAME=wmode VALUE=opaque>
     <PARAM NAME="AllowScriptAccess" VALUE="always"><EMBED
     src="//d2o307dm5mqftz.cloudfront.net/1505855001/1357341372265/Shipping_C.swf?' + flashVars + '" quality=high wmode=opaque
     swLiveConnect=TRUE WIDTH="300" HEIGHT="250" bgcolor=#FFFFFF TYPE="application/x-shockwave-flash"
     AllowScriptAccess="always"></EMBED></OBJECT>';
1288   document.write(adcode);
1289 }
1290 else {
1291   adcode = get3pPixed();
1292   adcode += '<A TARGET="_blank" HREF="' + clickURL + '"><IMG
     SRC="//d2o307dm5mqftz.cloudfront.net/1505855001/1357341372388/Shipping_C.jpg" alt="" width="300" height="250" BORDER=0>
     </A>';
1293   document.write(adcode);
1294 }
1295 </scripttag>
```

New Mitigation for Bitsquatting #2

- The majority of the bitsquat requests that Dinaburg received during his original bitsquatting research came from domain variants of Facebook's content delivery network fbcdn.net. Facebook is a web application.
- The web application design can be changed to help reduce the number of times the domain name appears in memory, thus reducing the number of opportunities for a bitsquat request.
- Using relative links inside of HTML instead of absolute links reduces the number of appearances of the domain name.
- With a base href, the domain name will appear at most once per HTML page. The downside is that if a bit error does occur in the base href, then all links in the document would go to the same bitsquat domain.

```
<!DOCTYPE html>
<html lang="en" id="facebook" class="no_js">
<head><meta charset="utf-8" /><script>function envFlush(a){function b(c){for(var d in a)c[d]=a[d];}if(window.requireLazy)
{requireLazy(['Env'],b);}else{Env=window.Env||
{};b(Env);}}envFlush({"user":"100001467532779","locale":"en_US","method":"GET","svn_rev":772429,"tier":"","push_phase":"V3
","pkg_cohort":"EXP1:DEFAULT","vip":"31.13.72.1","www_base":"https:\/\/www.facebook.com\/","rep_lag":2,"fb_dtsg":"AQA6c1fY
","ajaxpipe_token":"AXjknVkF0CH2lkXB","lhsh":"-
AQEWKrVB","tracking_domain":"https:\/\/pixel.facebook.com","retry_ajax_on_network_error":"1","fbid_emoticons":"1"});</scri
pt><script>envFlush({"eagleEyeConfig":{"seed":"0oHb","sessionStorage":true}});CavalryLogger=false;</script><noscript><meta
http-equiv="refresh" content="0; URL=/avc.tester?_fb_noscript=1" /></noscript><meta name="robots" content="noodp, noydir"
/><meta name="referrer" content="default" id="meta_referrer" /><meta name="description" content="Facebook is a social
utility that connects people with friends and others who work, study and live around them. People use Facebook to keep up
with friends, upload an unlimited number of photos, post links and videos, and learn more about the people they meet."
/><link rel="alternate" media="handheld" href="https://www.facebook.com/avc.tester" />
    <link rel="stylesheet" href="https://fbstatic-a.akamaihd.net/rsrc.php/v2/yp/r/CjUZmAGKPoZ.css" />
    <link rel="stylesheet" href="https://fbstatic-a.akamaihd.net/rsrc.php/v2/yh/r/K4pQGDu7_WJ.css" />
    <link rel="stylesheet" href="https://fbstatic-a.akamaihd.net/rsrc.php/v2/yd/r/_WELvqADscv.css" />
    <link rel="stylesheet" href="https://fbstatic-a.akamaihd.net/rsrc.php/v2/yb/r/_C2OlOkukPQ.css" />

    <script src="https://fbstatic-a.akamaihd.net/rsrc.php/v2/yR/r/YpD-WuoLxM8.js" crossorigin="anonymous"></script>
  <script>window.Bootloader && Bootloader.done(["6Ozhu"]);</script><script>Bootloader.loadEarlyResources({"kQ5UI":
{"type":"js","crossOrigin":1,"src":"https:\/\/fbstatic-a.akamaihd.net\/rsrc.php\/v2\/y-\/r\/lV3BV1YRc-7.js"},"hCTyG":
{"type":"js","crossOrigin":1,"src":"https:\/\/fbstatic-
a.akamaihd.net\/rsrc.php\/v2\/yH\/r\/OcdJkWzizD4.js"}});</script><script></script><title id="pageTitle">Avc
Tester</title><link rel="shortcut icon" href="https://fbstatic-a.akamaihd.net/rsrc.php/yP/r/Ivn-CVe5TGK.ico"
/><noscript><meta http-equiv="X-Frame-Options" content="deny" /></noscript>
    <link rel="stylesheet" href="https://fbstatic-a.akamaihd.net/rsrc.php/v2/yo/r/USyXIcPSwlv.css" />
    <link rel="stylesheet" href="https://fbstatic-a.akamaihd.net/rsrc.php/v2/yD/r/OWwnO_yMqhK.css" />
<script>new (require("ServerJS"))().handle({"require":[["removeArrayReduce"],["markJSEnabled"],["lowerDomain"],
["QuicklingPrelude"]]})</script></head><body class="_493u timelineLayout _51x9 _41h fbx webkit chrome mac
Locale_en_US"><div id="FB_HiddenContainer" style="position:absolute; top:-10000px; width:0px; height:0px;"></div><div
class="_li"><div id="pagelet_bluebar" data-referrer="pagelet_bluebar"><div id="blueBarHolder" class="slim"><div
id="blueBar" class="fixed_elem"><div id="pageHead" class="clearfix" role="banner"><h1 id="pageLogo"><a data-
gt="&#123;&quot;chrome_nav_item&quot;:&quot;logo_chrome&quot;&#125;" href="https://www.facebook.com/?
ref=logo">Facebook</a></h1><div id="jewelContainer" class="notifNegativeBase notifCentered notifGentleAppReceipt"><div
```

New Mitigation for Bitsquatting #3

- Capital ASCII characters are equivalent for DNS and URL hostname purposes, but possess fewer bitsquat variants
- There are no bitsquats of capital letters in the range 0-9
- The “.” is not a bit error variant of the capital letter “N”, only the lowercase “n”
- The “/” is not a bit error variant of the capital letter “O”, only the lowercase “o”
- The “#” is not a bit error variant of the capital letter “C”, only the lowercase “c”
- By simply substituting capital letters whenever lowercase letters “c” and “n” through “y” appear in a domain name, some bitsquat variants can be avoided.

Binary	Oct	Dec	Hex	Glyph	Binary	Oct	Dec	Hex	Glyph	Binary	Oct	Dec	Hex	Glyph
010 0000	040	32	20	␣	100 0000	100	64	40	@	110 0000	140	96	60	`
010 0001	041	33	21	!	100 0001	101	65	41	A	110 0001	141	97	61	a
010 0010	042	34	22	"	100 0010	102	66	42	B	110 0010	142	98	62	b
010 0011	043	35	23	#	100 0011	103	67	43	C	110 0011	143	99	63	c
010 0100	044	36	24	$	100 0100	104	68	44	D	110 0100	144	100	64	d
010 0101	045	37	25	%	100 0101	105	69	45	E	110 0101	145	101	65	e
010 0110	046	38	26	&	100 0110	106	70	46	F	110 0110	146	102	66	f
010 0111	047	39	27	'	100 0111	107	71	47	G	110 0111	147	103	67	g
010 1000	050	40	28	(	100 1000	110	72	48	H	110 1000	150	104	68	h
010 1001	051	41	29	)	100 1001	111	73	49	I	110 1001	151	105	69	i
010 1010	052	42	2A	*	100 1010	112	74	4A	J	110 1010	152	106	6A	j
010 1011	053	43	2B	+	100 1011	113	75	4B	K	110 1011	153	107	6B	k
010 1100	054	44	2C	,	100 1100	114	76	4C	L	110 1100	154	108	6C	l
010 1101	055	45	2D	-	100 1101	115	77	4D	M	110 1101	155	109	6D	m
010 1110	056	46	2E	.	100 1110	116	78	4E	N	110 1110	156	110	6E	n
010 1111	057	47	2F	/	100 1111	117	79	4F	O	110 1111	157	111	6F	o
011 0000	060	48	30	0	101 0000	120	80	50	P	111 0000	160	112	70	p
011 0001	061	49	31	1	101 0001	121	81	51	Q	111 0001	161	113	71	q
011 0010	062	50	32	2	101 0010	122	82	52	R	111 0010	162	114	72	r
011 0011	063	51	33	3	101 0011	123	83	53	S	111 0011	163	115	73	s
011 0100	064	52	34	4	101 0100	124	84	54	T	111 0100	164	116	74	t
011 0101	065	53	35	5	101 0101	125	85	55	U	111 0101	165	117	75	u
011 0110	066	54	36	6	101 0110	126	86	56	V	111 0110	166	118	76	v
011 0111	067	55	37	7	101 0111	127	87	57	W	111 0111	167	119	77	w
011 1000	070	56	38	8	101 1000	130	88	58	X	111 1000	170	120	78	x
011 1001	071	57	39	9	101 1001	131	89	59	Y	111 1001	171	121	79	y
011 1010	072	58	3A	:	101 1010	132	90	5A	Z	111 1010	172	122	7A	z

New Mitigation for Bitsquatting #4

- Create a RPZ containing bitsquats of popular or internal-only domains. These bitsquat domains can be configured with CNAMEs that point at the real domain, so your DNS resolver silently corrects bit errors without any work on the part of the client experiencing the bit error.
- There is a bitsquat domain of “paypal.com” called “raypal.com”. It is a real site, and not affiliated with PayPal, but it would be much more likely for a network’s users to be going to paypal.com instead. Therefore, some of these legitimate sites could either be whitelisted or just be counted as acceptable FPs.

Conclusion

- Bitsquatting is easier than it ever has been given the number of devices attached to the internet which lack error correcting memory. Bitsquatting will become easier to do over time.
- Bitsquatting affects many more domains than just the most popular. Less popular sites, and sites registered at “protected” TLDs like .gov, .edu, and .mil are vulnerable to some of these new techniques.
- Guarding against bitsquatting need not involve mass registration of domain names: Creative use of subdomains, use of relative hrefs in HTML, and use of href hostnames in CAPS can all help reduce the incidence of successful bitsquats. And since DNS is critical for bitsquatting attacks, using a DNS resolver with an RPZ that blocks/redirects likely bitsquat requests can provide the ultimate protection.

Special thanks to the following individuals. Without their assistance this research would not have been possible:

Thank you.

Seth Hanford
Adam Katz
Gavin Reid
Allyn Romanow
Henry Stern

Thank you.