


How A Large-Scale Russian Botnet Operation Stays Under the Radar

 blogs.infoblox.com/threat-intelligence/one-mikro-typo-how-a-simple-dns-misconfiguration-enables-malware-delivery-by-a-russian-botnet/

Infoblox Threat Intel

January 14, 2025



Author: David Brunsdon

Not too long ago Infoblox Threat Intel discovered a botnet delivering malware via spam campaigns using spoofed sender domains. This is different from the email spoofing that we recently reported on **Muddling Malspam: The Use of Spoofed Domains in Malicious Spam**, in that these take advantage of misconfigured DNS records to pass email protection techniques. Botnets, which are built out of actor-controlled compromised devices, are extremely difficult to disrupt and represent a persistent threat in the cyber landscape. This botnet uses a global network of Mikrotik routers to send malicious emails that are designed to appear to come from legitimate domains. The spam we observed delivered trojan malware, but the botnet is likely used for a wide range of malicious activities. We continue to track this botnet via DNS.

Some background on botnets

Botnets have existed since the early days of the internet and over time they have grown in sophistication. A botnet is a set of compromised devices that are remotely controlled by a threat actor to perform large-scale malicious actions ranging from spam distribution to denial-of-service attacks. The distributed nature of a botnet makes it difficult to disrupt and allows threat actors to hide their identities. Moreover, when an attack comes from thousands of sources like a botnet, it poses a much greater challenge for defenders to counter, versus an attack originating from a single IP address that can be easily blocked.

Typical uses for a malicious botnet include:

- Distributed denial of service (DDoS) attacks: Botnets can overwhelm a target's network or server with traffic, causing it to crash or become inaccessible. This can be used for extortion, to disrupt services, or as a distraction for other attacks.
- Spam and phishing campaigns: By using many compromised devices, attackers can send massive volumes of spam emails or phishing messages, increasing the likelihood of tricking recipients into revealing sensitive information or downloading malware.
- Credential stuffing: Botnets can automate the process of trying large numbers of username and password combinations on various websites, exploiting weak or reused credentials to gain unauthorized access.
- Data theft: Infected devices can be used to steal personal information, financial data, or intellectual property, which can then be sold on the Dark Web or used for further attacks.
- Cryptojacking: Botnets can hijack the processing power of infected devices to mine cryptocurrencies, generating revenue for the attackers without the victims' knowledge.
- Proxy networks: Compromised devices can be used as proxies to hide the attackers' true location, making it harder for law enforcement to trace their activities.
- Click fraud: Botnets can generate fake clicks on online advertisements, defrauding advertisers by making it appear as though their ads are receiving legitimate traffic.

An unexpected invoice

It all started with the discovery of a malspam campaign in late November. The email content was related to freight invoices and included a zip file containing a malicious payload. The zip files used a consistent naming convention of either:

- Invoice (2–3-digit number).zip or
- Tracking (2-3-digit number).zip

The emails indicate that the actor was impersonating the shipping company, DHL. Here is an example:

“

Invoice for Shipping Payment

Invoice Number: 728326122

Dear Customer,

We have prepared an invoice for the transportation of your package. Please find the attached file containing all the necessary payment information. To avoid delays in the delivery process, kindly make the payment as soon as possible.

Should you have any inquiries, please don't hesitate to reach out to us.

Best regards,

DHL Express

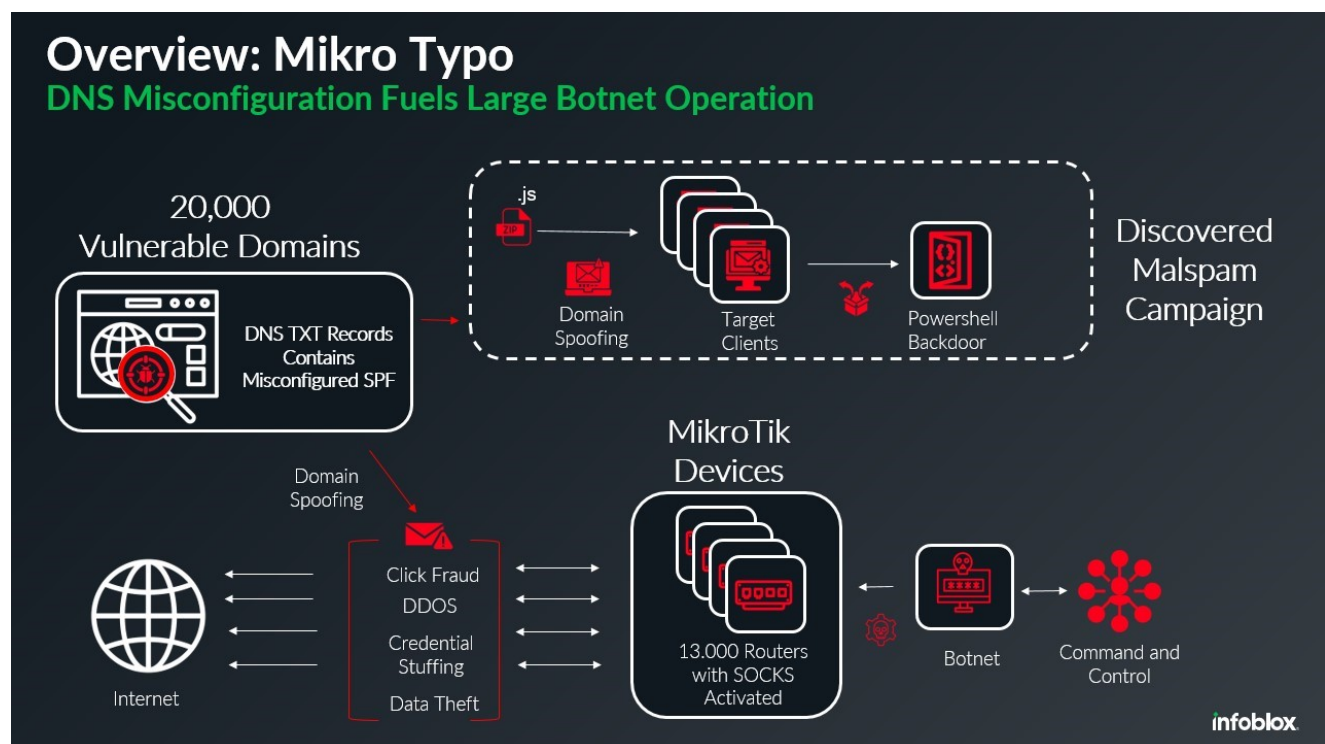
”

We compiled tens of thousands of these spam-designated emails and began our investigation.

The trojan

The attached zip file contains an obfuscated JavaScript file, which creates and executes a PowerShell script that initiates an outbound connection to the malware command and control (C2) server at 62.133.60[.]137. The IP, located on Global Connectivity Solutions (AS215540), has a history of suspicious use, related to prior Russian activity.

The botnet



How a misconfiguration in DNS enabled a botnet-powered malspam campaign

The headers of the many spam emails revealed a vast array of domains and SMTP server IP addresses, and we realized we had uncovered a sprawling network of approximately 13,000 hijacked MikroTik devices, all part of a sizeable botnet. Together, they form a large cannon, poised and ready to unleash a barrage of malicious activities.

There have been several critical vulnerabilities identified in MikroTik routers, and the firmware version of a router is not always available, but we saw a variety of versions impacted, including recent firmware releases. One remote code execution vulnerability with an exploit is described here:

<https://vulncheck.com/blog/mikrotik-foisted-revisited>

This buffer overflow exploit can be executed remotely but requires authenticated access. These routers, however, previously shipped with a hard-coded 'admin' user account with a blank password. (It would be good practice to disable this account). This vulnerability, and ones older, would not explain why we have seen recent firmware releases also enrolled as members of the botnet.

Regardless of how they've been compromised, it seems as though the actor has been placing a script onto the devices that enables SOCKS (Secure Sockets), which allow the devices to operate as TCP redirectors. Enabling SOCKS effectively turns each device into a proxy, masking the true origin of malicious traffic and making it harder to trace back to the source. Another significant concern is that the lack of authentication required to use these proxies makes individual devices, or the entire botnet, available for other actors to exploit. Even though the botnet consists of 13,000 devices, their configuration as SOCKS proxies allows tens or even hundreds of thousands of compromised machines to use them for network access, significantly amplifying the potential scale and impact of the botnet's operations.

The broad implications of having this many routers and domains accessible for abuse are troubling. With such a large network at their disposal, an actor can launch a variety of attacks, from DDoS assaults to more covert operations like data exfiltration and phishing campaigns. Each compromised device becomes a cog in a much larger, nefarious machine, capable of wreaking havoc on unsuspecting targets.

Moreover, threat actors can use SOCKS proxies to spread malware by routing their traffic through these compromised devices, bypassing traditional security measures. An operator of command and control (C2) infrastructure with access to this botnet could greatly increase the scale of their operations and it would also provide a layer of anonymity, shielding them from detection and blocking.

Where does DNS come in?

For a malspam threat actor, assuring your malicious payloads get past the protections in place to prevent spam, and are forwarded onto potential victims, is a critical task. Mail servers use several types of DNS TXT records, including DomainKey Identified Mail (DKIM) and Sender Policy Framework (SPF), and Domain-based Message Authentication, Reporting, and Conformance (DMARC), to handle email, and in general, do a terrific job of preventing spam. In this case, however, a misconfiguration in the SPF records of domains gave the malspam actors an opportunity to break through these protections.

When a user sends an email, the receiving mail server checks the SPF record to verify that the message is coming from an authorized server. If the email fails this check, it is more likely to be marked as spam or rejected. The SPF information is published in the domain's DNS records as a TXT record.

The malspam campaign we investigated was large in scope, involving approximately 20,000 sender domains. Although the domain owners configured SPF, they were configured such that any address can send emails for their domains.

This DNS misconfiguration could have been done by accident, or as a malicious modification by a threat actor with access to the domain's registrar account. Either way, the consequence is that any device can spoof the legitimate domain in email.

Here's an example of a properly configured SPF record for example.com. It identifies a specific server that is valid to send emails of the domain, and says to deny any other:

```
v=spf1 include:example.com -all
```

In this example, only the servers specified by example.com are authorized to send emails for example.com. The "-all" at the end means that any other server attempting to send emails on behalf of example.com will be denied. However, the "~all" option, which denotes a 'soft fail,' is more commonly used. This option allows the receiving mail servers to make their own handling decisions, such as marking the email as suspicious or delivering it with a warning.

And here's an example of the kind of misconfigured SPF record we observed in this campaign:

```
v=spf1 include:example.com +all
```

In this case, the servers specified by example.com are included, but the "+all" at the end means that any server is allowed to send emails on behalf of example.com. This essentially defeats the purpose of having an SPF record, because it opens the door for spoofing and unauthorized email sending.

The self-checkup

If you've got a domain name that you want to check to see if it has an SPF record and how it's configured, just look in its DNS TXT records.

On Linux or MacOS we can check 'example.com' using dig with grep:

```
dig +short txt example.com | grep spf
```

On Windows we can use nslookup with a bit of PowerShell.

```
nslookup -type=txt example.com | Select-String -Pattern "spf"
```

In either example, we get this result:

```
"v=spf1 -all"
```

And we can see that 'example.com' isn't allowing mail from their domain at all.

Conclusion

Through mapping out the activity of this botnet related to the malspam campaign, we identified over 13,000 compromised MikroTik devices and 20,000 domains involved in sending spoofed mail. We also identified that these devices have been compromised in a way that allows them to be operated as an open (SOCKS4) relay.

This botnet is a stark reminder of the evolving threats in the cybersecurity landscape. With so many compromised MikroTik devices, the botnet is capable of launching a wide range of malicious activities, from DDoS attacks to data theft and phishing campaigns. The use of SOCKS4 proxies further complicates detection and mitigation efforts, highlighting the need for robust security measures.

The malspam campaign that led to this discovery exploited misconfigurations in DNS SPF records, allowing the threat actor to bypass traditional email protection measures. This underscores the importance of proper DNS configurations and regular audits of security settings, including the accessibility of your devices to the outside world, to prevent such vulnerabilities.