LIFARS
your digital world, **secured**

PHISHING INFRASTRUCTURE

Phishing Capabilities Demonstration

April 2021

TABLE OF CONTENTS

LIFARS
your digital world, secured
244 Fifth Avenue, Suite 2035, New York, NY 10001
LIFARS.com (212) 222-7061 info@lifars.com

INTRODUCTION

The goal of this whitepaper is to summarize technical details of a phishing infrastructure we developed and has unique capabilities among its open-source alternatives. It is capable of sending phishing emails using multiple techniques, bypassing multi-factor authentication and consequently stealing users' logged in sessions, with a possibility of a manual takeover or doing automated actions.

Our goal is to demonstrate capabilities a dedicated attacker might create using off-the-shelf open-source tools and some programming effort in putting them together. This article does not aim to provide a step-by-step solution to build a phishing infrastructure.

Phishing

Phishing email is a type of online scam when a cyber-criminal sends an email that appears legitimate but tricks the user into doing some actions that help the attacker. Most common is credential harvesting, which tricks a victim into disclosing their username and password, or they are tricked into running an executable, which can give the attacker access to the computer.

According to IBM[1], phishing still lies at the root of 14% of all data breaches, making it the 4th most used attack vector. Verizon Data Breach 2020[2] estimates, that 22% of all data breaches involved phishing. With the average data breach costing as much as $3.86 billion, no avenue can be left unchecked. Even tech giants such as Facebook and Google have fallen victim to multi-million dollar phishing scams. Although many security solutions have built-in phishing detection and prevention tools, attackers are continuously discovering new techniques for phishing filter evasion.

Multi-factor authentication

Multi-factor authentication or MFA is an authentication method, which requires two or more verification factors to grant access to the user. These factors can be categorized as either knowledge (something only the user knows), possession (something only the user has), or inherence (something only the user is). The best policy is to require at least two factors from different categories.

A typical example is an authentication with a password and a time-based one-time password (TOTP) generated by an application on a phone only the user has.

MFA can mitigate against many kinds of attacks, for example an attacker can no longer gain access to a service by only knowing user's credentials. Similarly, a simple credential harvesting phishing loses its efficacy.

However, attackers are always looking for a new way to bypass restrictions, and they have a capability to do phishing, which bypasses MFA[3]. We will demonstrate how such infrastructure could work in this document.

[1] https://www.ibm.com/security/digital-assets/cost-data-breach-report/
[2] https://enterprise.verizon.com/resources/reports/2020-data-breach-investigations-report.pdf
[3] https://info.publicintelligence.net/FBI-CircumventingMultiFactorAuthentication.pdf

LIFARS your digital world, secured
244 Fifth Avenue, Suite 2035, New York, NY 10001
LIFARS.com (212) 222-7061 info@lifars.com

COMPONENTS

In this chapter we introduce each component of the infrastructure separately, before they are put together. The infrastructure should be capable of:

- Sending a phishing email to a huge number of recipients, with support of templating
- Track which users opened a phishing email, clicked on a link and submitted data
- Bypass MFA by using real-time phishing
- Automatically carry out actions with phished sessions
- All components should communicate together with the least amount of action by operator
- Make reporting of results as easy as possible

Gophish

Gophish[4] is a powerful open-source phishing framework, which makes it simple to manage email templates, addresses of email recipients, email addresses from which the email is sent as well as monitor how the phishing campaign is going.

Main reasons why it was chosen as a component are: a practical UI, a relative software maturity and API which can and will be used for cooperation with other tools. However, out-of-the-box it only supports a simple credential harvesting attack, which is less practical with the advent of MFA.

Multiple ways to send the mail

The "Sending Profiles" functionality allows users to set up multiple ways of sending an email and have them ready for sending. There are multiple ways that this can be set up, not just sending an email directly to a victim from Gophish.

Marketing email services

Attackers are known to use legitimate marketing email services, such as SendGrid[5] or MailChimp[6] to send their phishing, since it allows them to send the email from IP addresses that are known to send some legitimate messages.

Most of these services allow for emails to be sent through SMTP with some specific username and password, which means it can be easily set up as a sending profile in Gophish.

[4] https://getgophish.com/
[5] https://sendgrid.com/
[6] https://mailchimp.com/

LIFARS your digital world, secured
244 Fifth Avenue, Suite 2035, New York, NY 10001
LIFARS.com (212) 222-7061 info@lifars.com

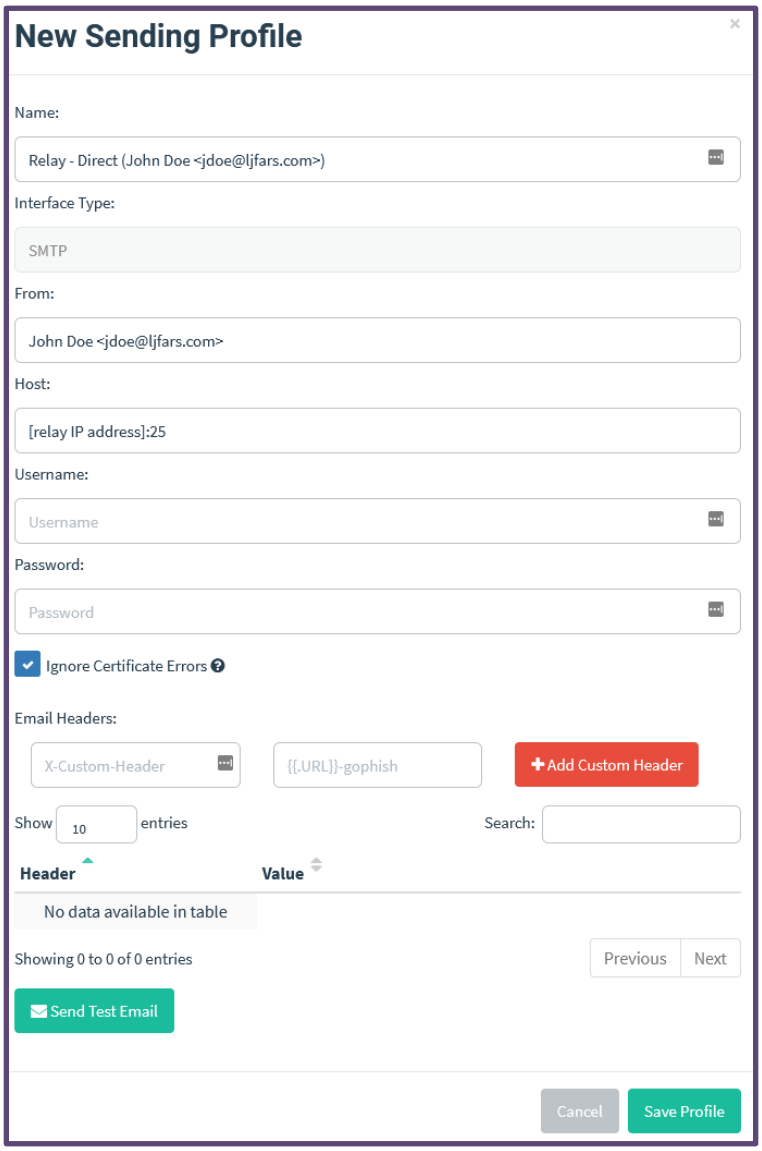

Figure 1 Example of a Sending Profile configuration in Gophish

Relay server

The phishing emails can be sent indirectly, using another server which would receive all the messages from Gophish and send them to their intended recipients. If its IP address gets compromised or gets a bad spam score, it is significantly easier to set up the same SMTP server on a different IP address, than to move the entire infrastructure. This resender should also strip all non-essential headers and by doing so, mask the original infrastructure.

One downside of this approach, which is also present when sending the emails directly is, that full responsibility for reputation of the IP address is on the operator. The setup needs to gain a good reputation almost from scratch in relatively short time in a phishing engagement. Few positive points can always be earned by correctly setting SPF, DKIM and DMARC records for a phishing domain.

LIFARS your digital world, secured
244 Fifth Avenue, Suite 2035, New York, NY 10001
LIFARS.com (212) 222-7061 info@lifars.com

Resending the emails can be done for example by Postfix[7] or hMailServer[8]. Postfix can be configured in a way, that it resends all the emails from a specific Gophish IP address to their intended recipients, while also stripping all traces which could lead back to the original server. Furthermore, it can be configured so if an email comes from a specific email account, it can be forwarded to one of marketing services mentioned previously.

For a use-case, when a similar looking domain is bought, for which it would be too cumbersome to configure a whole mail server, this mail relay can also be instructed to forward all emails received for these domains to a specific mail account.

Gophish Alternatives

- Phishing Frenzy[9]
- King Phisher[10]
- LUCY[11] (paid tool, which has free version with basic functionality)

Evilginx2

Evilginx2[12] is a man-in-the-middle framework for harvesting credentials, as well as 2nd factor data, and by its use the infrastructure is able to bypass most types of MFA, steal users' sessions and therefore use the application a user logged into as the legitimate user would.

When deployed, it acts as a reverse proxy for the legitimate website. It resends every request it gets to the original website and sends the response back to the victim.

We will refer to the Evilginx2 webserver as `attacker.com` and to the legitimate login `legitimate-login.com`. The process is also described in Figure 2 Diagram of successful phishing through Evilginx2.

In the first step a victim is persuaded by a phishing email to visit `attacker.com`. When their browser makes an HTTP request to `attacker.com`, Evilginx2 receives it and makes the same request, but to `legitimate-login.com`. Evilginx2 gets an HTML code of `legitimate-login.com` and modifies it – all the links are changed to a subdomain of `attacker.com`, so following requests from the victim are sent to Evilginx2. JavaScript functionality can be changed as well. This edited HTML code is then sent by Evilginx2 to the victim as a response from `attacker.com`.

If the victim submits their login credentials to `attacker.com`, Evilginx2 saves them and sends them to `legitimate-login.com`, same as it did with requests before. The legitimate login page responds with HTML code for 2FA prompt and Evilginx2 rewrites all the links, so they point to Evilginx2 and sends it to the victim as a response from `attacker.com`.

[7] https://www.postfix.org/
[8] https://www.hmailserver.com/
[9] https://github.com/pentestgeek/phishing-frenzy
[10] https://github.com/rsmusllp/king-phisher
[11] https://lucysecurity.com
[12] https://github.com/kgretzky/evilginx2

LIFARS your digital world, secured
244 Fifth Avenue, Suite 2035, New York, NY 10001
LIFARS.com (212) 222-7061 info@lifars.com

After the victim submits correct MFA token, it is again forwarded by Evilginx2 to `legitimate-login.com`, however, when the login website responds with a successful login, Evilginx2 recognizes that and redirects the victim to an attacker-chosen website. It is configured to something user expects in the phishing scenario, to alleviate any suspicion. Evilginx2 saves the session token that was transmitted with the successful login.

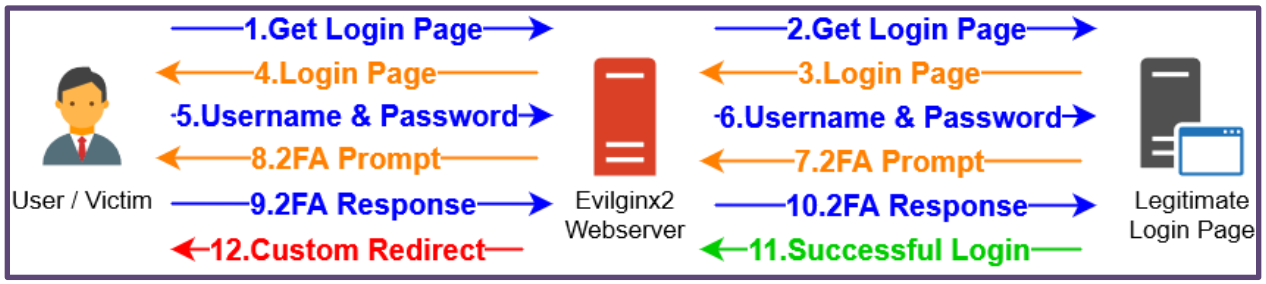

Figure 2 Diagram of successful phishing through Evilginx2

This example demonstrated what happens if MFA uses TOTP generated by either user's application or user's physical device. This technique is effective even against other forms of MFA, like a prompt to accept or decline login sent to an application user has on their second device. In that case, user would not receive a 2FA prompt as is displayed in Figure 2, they would only receive a notice that a request was sent to their second device, and if it is accepted, a redirection signifying a successful login. In other words, `attacker.com` would behave exactly the same as `legitimate-login.com`, and if users are used to this form of MFA, there is a good chance they would accept it.

From a point of view of `legitimate-login.com`, user successfully logged in from the Evilginx2 server. This means, that if an attacker makes any action from IP address of the Evilginx2 server with correct session tokens and the same user-agent, it should be no different than if the user made them.

Phishlet creation

For Evilginx2 to recognize which URLs to rewrite from `legitimate-login.com` to `attacker.com`, which cookies are session tokens and what is the legitimate login page, it needs to be configured. This takes place in so-called Phishlets.

A Phishlet is a configuration file in yaml format, which specifies a behavior for Evilginx2. It is usually made specifically for a service which should be mimicked. It contains a list of domain names which should be impersonated, for example that `account.legitimate-login.com` should be rewritten to `acc.attacker.com` and every request sent to `acc.attacker.com` should be sent to `account.legitimate-login.com`. A script can also be inserted, potentially changing any aspect of the website.

Evilginx2 has a library of ready-made Phishlets for many services, that usually work out-of-the-box, even though some require a degree of tweaking. However, to make a Phishlet from scratch, particularly to find what requests are made for MFA, one ideally needs a testing account or a reference infrastructure, which would use the same mechanisms. On the other hand, a Phishlet which only does credential harvesting can be always tested, even without a valid credentials.

Another great feature of Evilginx2 is, that it allows an arbitrary rewrite of the webpages that it relays. For example, if a webpage has a JavaScript check if it is on a correct domain, this can be removed before it reaches the victim's browser.

Evilginx2 Alternatives

- Modlishka[13]
    - Similar to Evilginx2
- Muraena[14]
    - With NecroBrowser[15] it should be able to replace our "autonomous scripts"
    - There is not much documentation available
    - Probably needs code analysis to configure properly
- ReelPhish[16]
    - No new commits since 2018
    - Does not function as a proxy
        - Phishing site has to be made from scratch
        - Login sequence has to be configured or programmed

Autonomous scripts

The sessions, that are stolen by Evilginx2 can be used manually, however if the infrastructure is not monitored constantly, the sessions might expire or not be used at all because of amount of tedious work it would entail.

Autonomous scripts were created to automate this part of an engagement. They are short scripts, which instruct headless browser, Selenium[17] in this case, to carry out certain actions autonomously, using the sessions stolen by Evilginx2.

These scripts can, for example, take a screenshot of a logged-in user and harvest some data as a proof in phishing engagements, or regularly visit the website just so the session does not expire due to inaction and it is still available for manual takeover.

Attackers could gain persistence by setting up a new multi-factor token, or issuing an application password. It is also possible to do application-dependent actions, such as dump all user emails or send another wave of phishing as the victim, boosting its credibility.

There is no limit to what actions can be done automatically and autonomously, unless the functionality is protected by CAPTCHA or requires another authentication with MFA. Keep in mind, that if there was a successful login, the script should be able to use correct username and password, since Evilginx2 saved them.

[13] https://github.com/drk1wi/Modlishka
[14] https://github.com/muraenateam/muraena
[15] https://github.com/muraenateam/necrobrowser
[16] https://github.com/fireeye/ReelPhish
[17] https://www.selenium.dev/

LIFARS your digital world, secured
244 Fifth Avenue, Suite 2035, New York, NY 10001
LIFARS.com (212) 222-7061 info@lifars.com

INTERCONNECTIONS AND IMPROVEMENTS

This chapter describes how the components fit together and what changes we made to some of the tools to improve their cooperation, overcome some of their shortcomings or improve them.

Sum of all parts

The infrastructure consists of following components:

- Gophish v0.11.0 – Store for recipient information and sending profiles, phishing emails creation and sending, data visualization
- Evilginx2 v2.4.0 – A phishing page where users are redirected, feeds data back to Gophish
- Autonomous scripts – Do predefined actions with the session tokens from Evilginx2
- (Optional) Proxies/Redirectors – redirect traffic from/to the phishing infrastructure to mask it from the outside world and make traffic less geographically suspicious

Transparency headers

Both Evilginx2 and Gophish contain HTTP or SMTP headers to identify themselves and give defenders means to pinpoint what tools are the attackers using. Even though this is a useful feature for some phishing engagements, it could be detrimental to red teaming, so we modified the code to get rid of them.

Evilginx2 logging

The downside of Evilginx2 in version 2.4.0 we used is, that it does not log into files. If it crashed and database got corrupted, one may lose the captured data.

A quick workaround is to run script[18] command before running Evilginx2, for example, `script phishing.log` to save all the output to a file phishing.log. Evilginx2 code was also modified, so it logs the most important events into a separate file, for example when a new user arrives, credentials they submit and the session tokens.

Gophish and Evilginx2 interaction

These two tools are the crux of our infrastructure and their cooperation is essential. There are multiple instances where we had to come up with some solutions that would allow them to work together.

Campaign creation

In a typical deployment, Gophish sends a link pointing to itself, so when a victim clicks the link in the email, it is redirected to the Gophish landing page. In this case however, it needs to redirect the user to the Evilginx2 site. Furthermore, to track which user clicked which link, they need to be unique.

[18] https://man7.org/linux/man-pages/man1/script.1.html

Luckily, Evilginx2 has a functionality, that can store an arbitrary key/value pair in URL, which is deciphered on the server when it is clicked.

We created a script, which uses Gophish API to create a CSV file with all the user email addresses for a given user group, then the operator manually executes a command in the Evilginx2 menu, that loads aforementioned CSV file, encodes the email address for each user into a unique URL link and outputs another CSV file with the unique URL for each email. Then, Gophish API is used once more to load these URLs into users' Position variable.

When creating a template in Gophish, one does not use `{{.URL}}` to specify a link address, but `{{.Position}}`. Also, the variable Position cannot be used in Gophish anymore, however it seems to be a worthwhile trade.

Opened email tracking

After all of the links were redirected to Evilginx2 instead of Gophish, it lost a significant feature – tracking opened emails. That feature in Gophish works by embedding an invisible picture in an HTML email, with URL like `https://attacker.com/track?rid=tNXICsO` where `rid` parameter is unique for each user. When this URL is visited, Gophish marks the user as one that opened a phishing email.

This functionality could be recreated by either building in a logic in Evilginx2, that visits the Gophish link with the correct rid parameter, or by setting Gophish up on a different domain or a different port and pointing the tracking picture link there.

In this infrastructure, Gophish and Evilginx2 are installed and run on the same server, and it is intended for them to use the same hostname. We configured the landing page for Gophish to reside on another port, for example 8080, and set up the campaign to have the tracking link on the same hostname, but different port. Gophish can even be configured to use the TLS certificates, which were automatically created by Evilginx2.

Clicked link and password submission tracking

Another functionality which has to be recreated is tracking which users clicked on the phishing link. While it can done manually by parsing out unique clicked links from Evilginx2 logs and cross-referencing them back to the usernames or emails, it could be done automatically and use visualization power of Gophish.

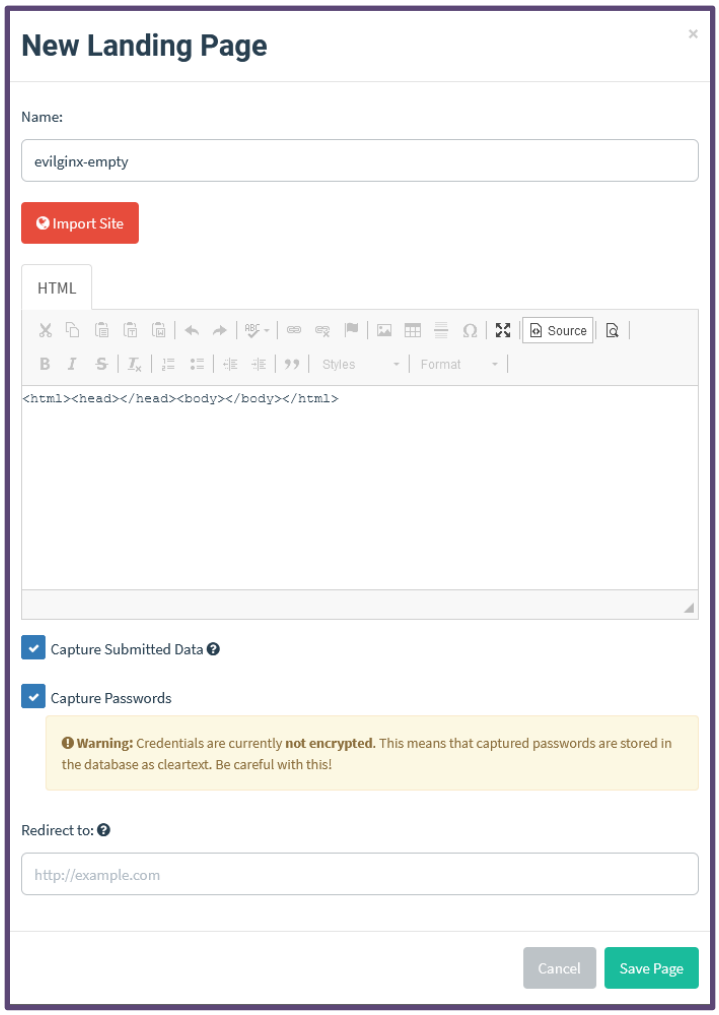

Figure 3 Example of empty landing page configuration in Gophish

We built a script, which fetches a `rid` parameter for each user in a given campaign through Gophish API and saves it in a file. We also modified the Evilginx2 code in a way, that it looks for the correct `rid` parameter in this file when a visited link is successfully paired to a user email. Evilginx2 then makes a request for the Gophish landing page, with the correct `rid` parameter, same User-Agent that it received from the victim and puts the victim's IP address in `X-Forwarded-For` HTTP header. From the point of view of Gophish, all of the data is legitimate and the statistics it creates are completely sound.

For the password submission, Gophish accepts any data in an HTTP POST request with parameters `username` and `password`. We modified Evilginx2 to first find the correct `rid` in the same way as for clicked link tracking, and then to submit the data using a simple HTTP request. When configuring an empty landing page in Gophish, the `Capture Passwords` option has to be selected.

Evilginx2 and autonomous scripts interaction

Scripts, that carry out actions while impersonating a user need data from Evilginx2, like the session tokens and a User-Agent value. This information is kept in database file `~/.evilginx.data.db`, which is written using BuntDB[19] library in Golang.

We created a short Golang script using the same library, which takes the User-Agent and the cookie data from the database and writes them into a JSON file, which can then be read and used by a script in almost any language.

These scripts need to be triggered and run automatically, which requires more interaction with Evilginx2. There are multiple ways to doit. The simplest solution is to run the script in a loop with some timeout. Another way is to use common commands like inotifywait[20] to watch Evilginx2 database and when it changes, runs the script. Most laborious way is to modify Evilginx2 to run the script, for example, when a new session is captured.

[19] https://github.com/tidwall/buntdb
[20] https://linux.die.net/man/1/inotifywait

DEMONSTRATION

We demonstrate a successful phishing attack on Outlook webmail protected by the built-in 2FA protection.

First, we buy a lookalike domain, in this case `ljfars.com`, and set the DNS record for all the subdomains to the IP address of our Evilginx2 server. If Evilginx2 does not have a TLS certificate for the domain it is configured to serve, it requests it from Let's Encrypt[21], which requires DNS to be set up. This certificate can be then used by Gophish as well.

We use a pre-made Evilginx2 Phishlet and put email recipients into a new Gophish group. Our custom script is run to generate Evilginx2 links for each user and load them through Gophish API into "Position" column. That changes the data in "Position" column to a unique link for each recipient.

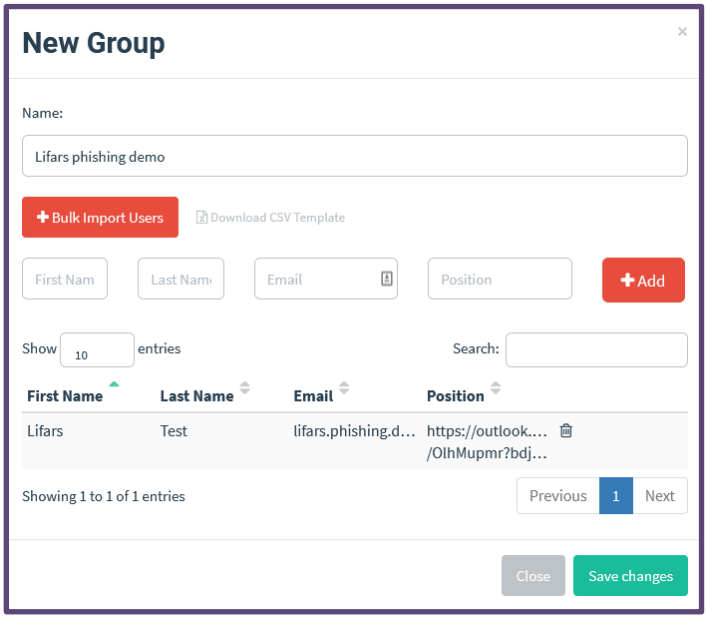

Figure 4 Gophish group with link in "Position" column

An email template is created, we use `{{.Position}}` as a placeholder where a link should go. When sending phishing emails, Gophish replaces `{{.Position}}` with a record in "Position" column for each user, which in this configuration is their unique link for phishing.

[21] https://letsencrypt.org/

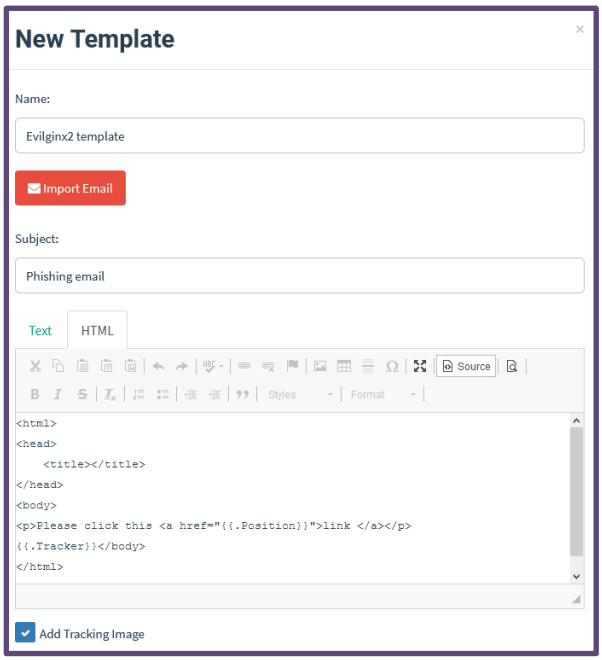

Figure 5 Phishing email template

We create a new campaign in Gophish, select a premade email template, a group of recipients, an empty landing page and a sending profile, which will send the phishing directly, without a relay or a marketing email service and then launch the campaign.

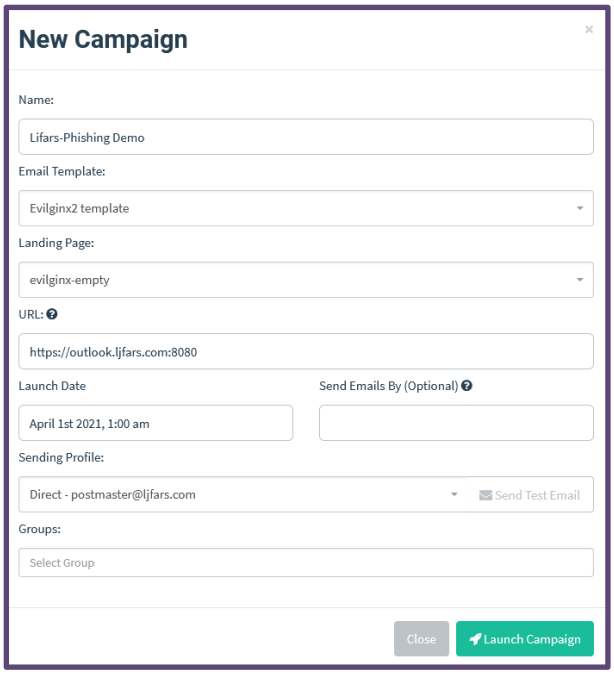

Figure 6 Campaign configuration

LIFARS your digital world, secured
244 Fifth Avenue, Suite 2035, New York, NY 10001
LIFARS.com (212) 222-7061 info@lifars.com

When the campaign is launched, we need to transfer unique Gophish `rid` identifiers to Evilginx2 using our custom script.

```
pt@guppy-1:~/.evilginx$ ./gophish_getrid.py
68      tracking test
69      tracking test 2
70      Lifars Phishing Demo
Select ids of campaigns you want to get rids from, comma delimited:70
1 RIDs written to file rids.csv
```

Figure 7 Transferring "rid" parameters from Gophish to Evilginx2

Victim receives and opens the phishing email, which prompts a request for the invisible tracking picture and Gophish is notified that the message was opened.

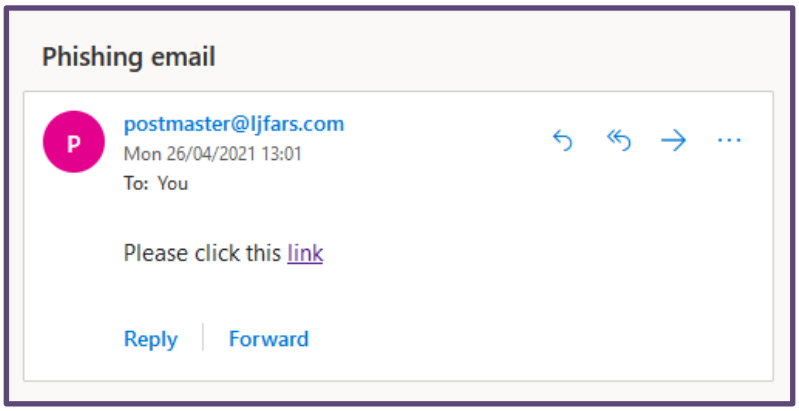

Figure 8 Received email

LIFARS your digital world, secured
244 Fifth Avenue, Suite 2035, New York, NY 10001
LIFARS.com (212) 222-7061 info@lifars.com

The victim clicks the link and is presented with a login screen, which looks identical to the legitimate login screen, it is just hosted on `login.ljfars.com` instead of `login.live.com`.

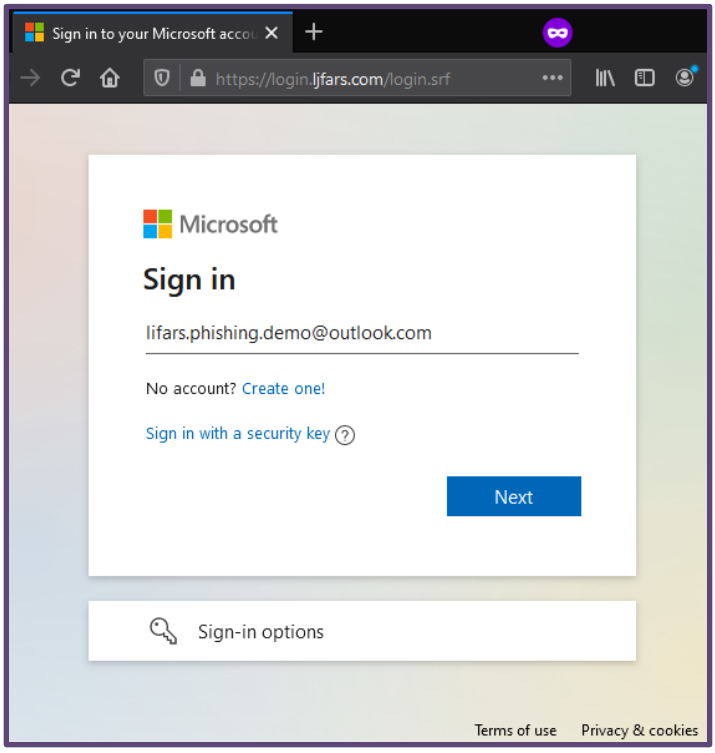

Figure 9 Spoofed login screen

Victim submits their username and password, which are captured and saved by Evilginx2 and forwarded to Gophish. Since Evilginx2 just relays the data, if the victim submits incorrect credentials, an error page is shown in the same way as it would be on the real login page.

LIFARS your digital world, secured
244 Fifth Avenue, Suite 2035, New York, NY 10001
LIFARS.com (212) 222-7061 info@lifars.com

Credentials were correct, so the victim is prompted for their 2FA token, in this case TOTP in a form of numeric code, which is generated in Authenticator app.

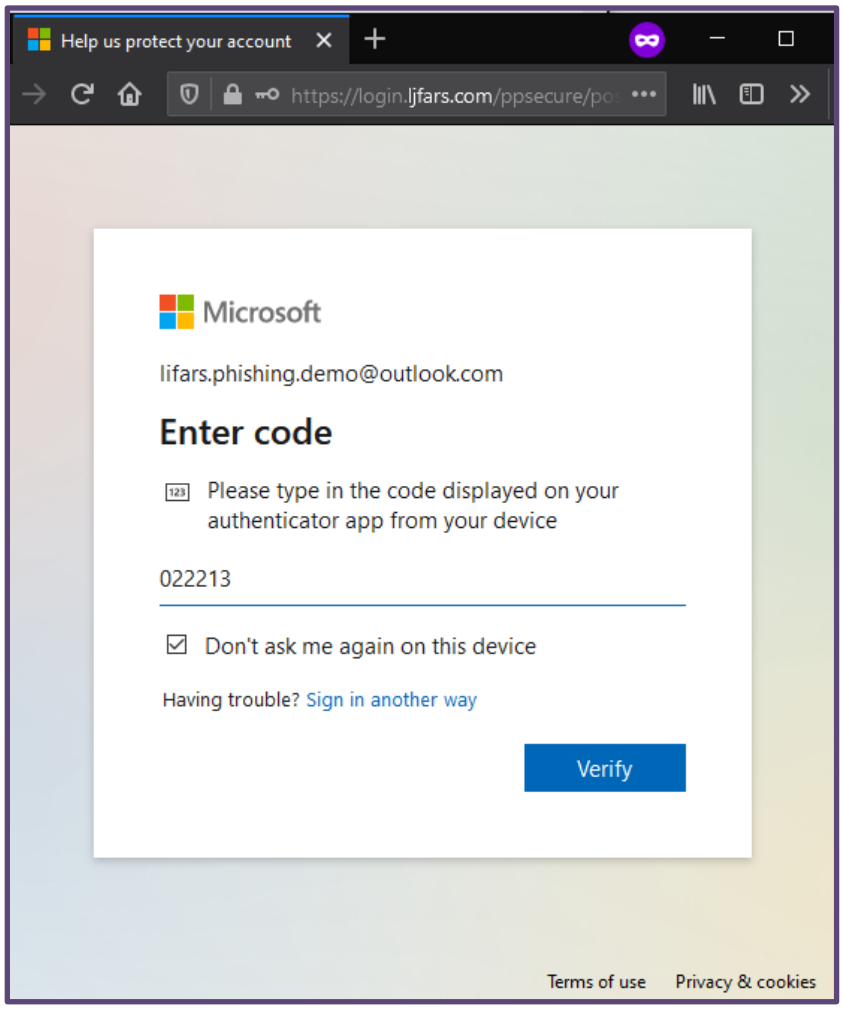

Figure 10 Time-based One-Time Password page

If the numeric code is correct, Evilginx2 captures session tokens and redirects the victim to an arbitrary site chosen by the attacker. It should be something that maintains the illusion which was created by the phishing email and does not raise suspicion.

LIFARS your digital world, secured
244 Fifth Avenue, Suite 2035, New York, NY 10001
LIFARS.com (212) 222-7061 info@lifars.com

From the attacker's point of view, they can see a victim clicked a link and submitted their password in Gophish. Events are arranged into a timeline with timestamps. They can also see what operating system and browser the victim uses, which is parsed from the User-Agent value. Gophish also has the data about IP address the request made from.

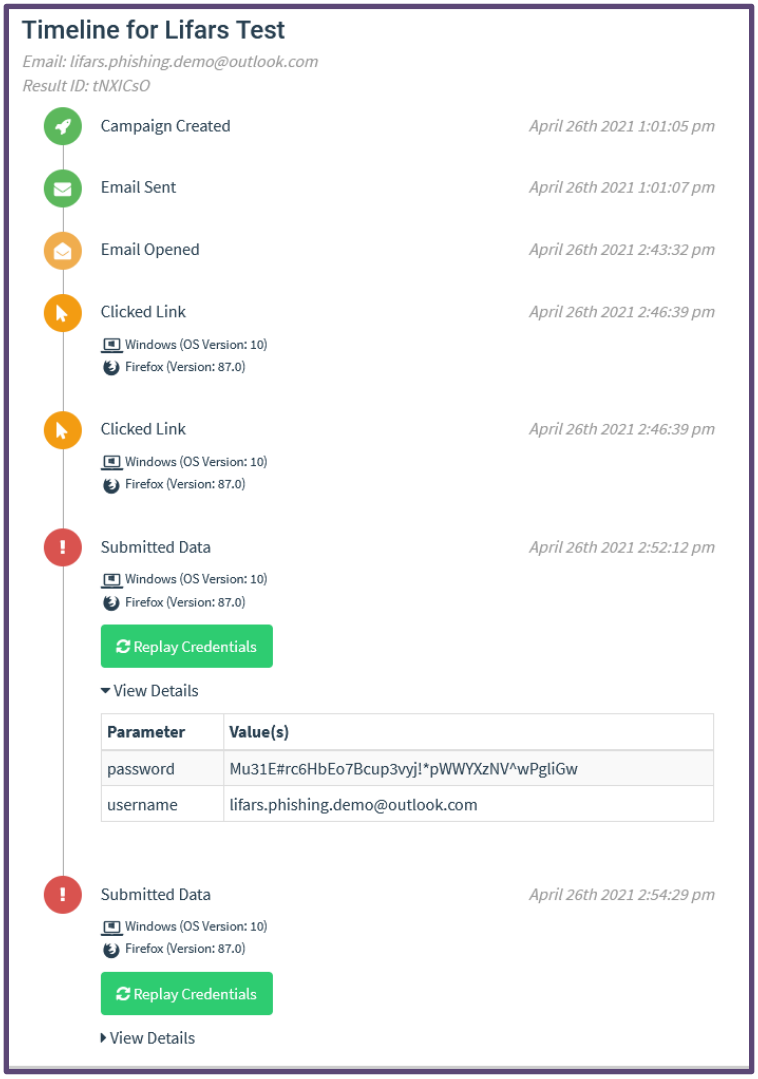

Figure 11 Gophish timeline

LIFARS your digital world, secured
244 Fifth Avenue, Suite 2035, New York, NY 10001
LIFARS.com (212) 222-7061 info@lifars.com

The attacker can look up the details about this session in Evilginx2 command line as well. It also contains data about the source IP address, the User-Agent value, the username, the password, timestamps and most importantly – the session tokens, which can be used to hijack the user session.

Figure 12 Details about the captured session in Evilginx2

The attacker uses a browser, which forwards all of its traffic through the Evilginx2 server, so from the point of view of the legitimate login server, nothing is suspicious. They set these cookies in the browser, for example by using Cookie Editor.

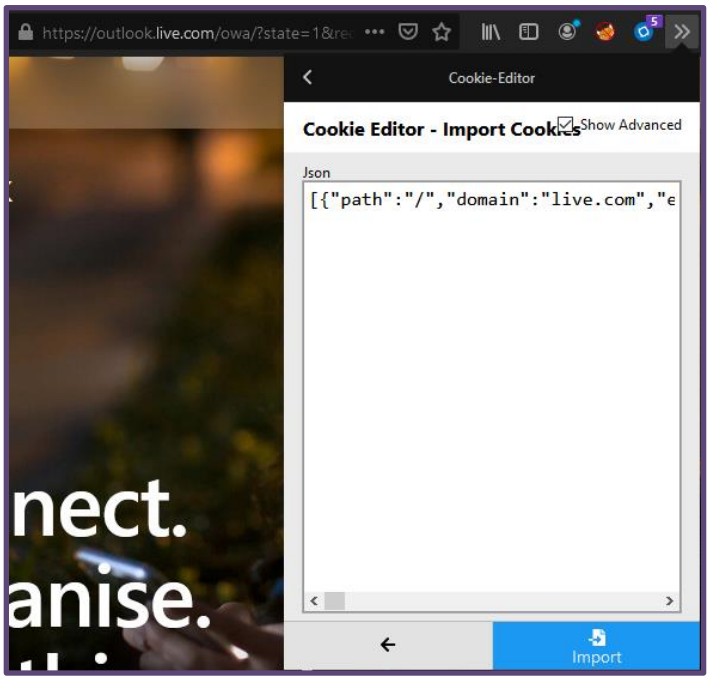

Figure 13 Setting cookies in the browser

LIFARS your digital world, secured
244 Fifth Avenue, Suite 2035, New York, NY 10001
LIFARS.com (212) 222-7061 info@lifars.com

Then they visit `outlook.live.com` and are successfully logged-in as the victim. They can interact with the application as a normal user – they can read, move and delete emails, change settings or send new emails to further their phishing campaign. Furthermore, any action that can be done manually can also be done automatically by the automated scripts, as long as it does not require CAPTCHA or another authentication with MFA.

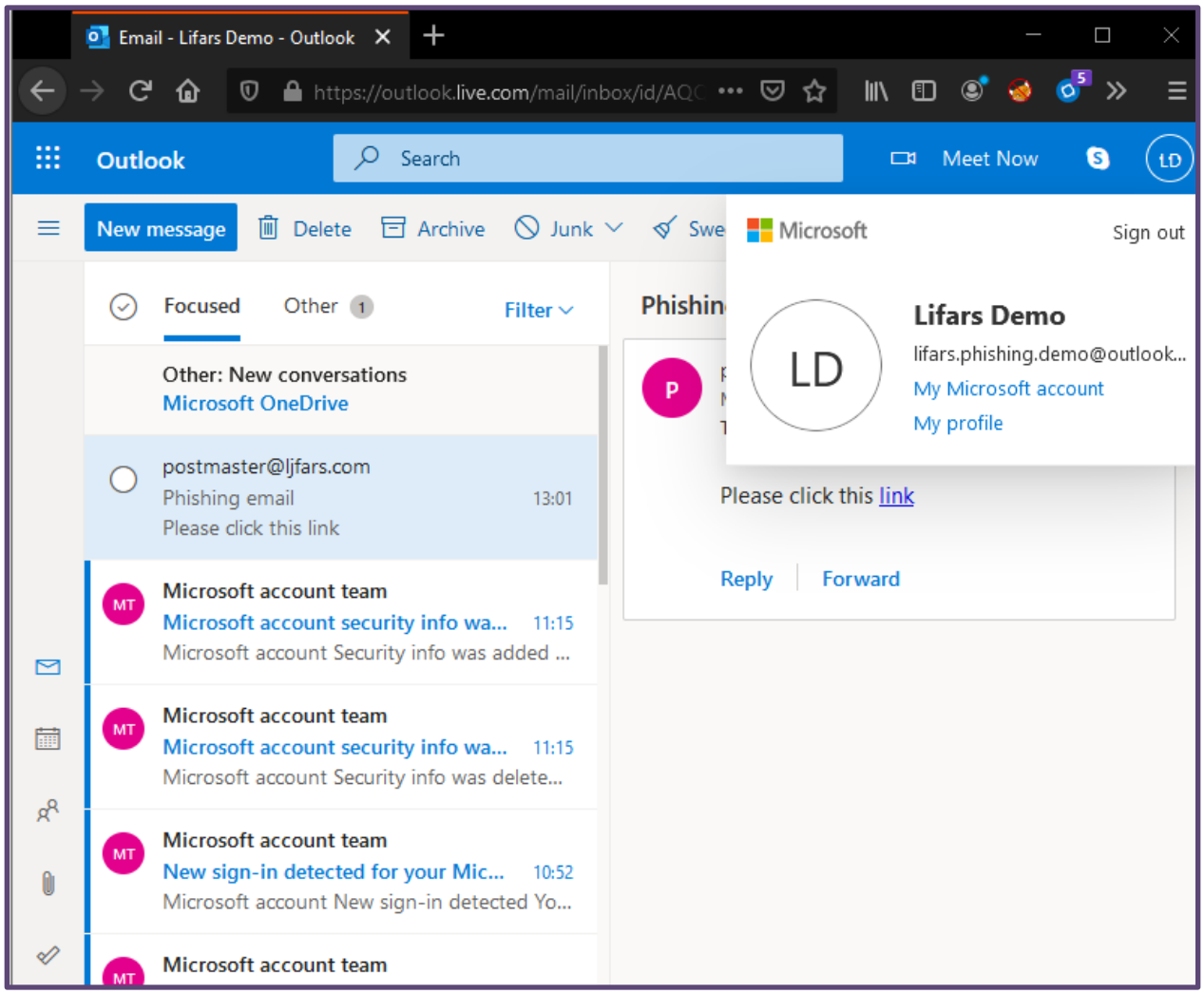

Figure 14 Login by attacker after successful phishing

In this simple demonstration we presented how an attacker could hijack a session for an Outlook webmail. However, other web applications protected by MFA can be targeted as well. If the attack succeeds on a Single-sign-on service like Okta, attacker may gain access to all of the applications in an organization.

LIFARS
your digital world, secured
244 Fifth Avenue, Suite 2035, New York, NY 10001
LIFARS.com (212) 222-7061 info@lifars.com

MITIGATIONS

The typical defenses against phishing are effective for this one as well. Attack would be foiled if the users noticed that the login page is not hosted on its usual hostname. Impersonating the legitimate login page with the same URL through functioning HTTPS protocol with a valid TLS certificate should be very hard for attackers to do. Also, when sending the phishing emails, they might be caught by anti-spam solution. In these aspects, the attack is not any different from a typical phishing.

Since these attacks usually use similar-looking domains, it is wise for administrators to either buy or monitor them, for example by using periodically running the dnstwist[22] tool or other services that do brand monitoring. Some modern browsers like Chrome already have a protection against them – when a user has a long established history of visiting specific hostname, requesting a similar-looking hostname will alert the user with message "Did you mean [hostname]?", which may alert some users.

The demonstrated technique would also not work for MFA which uses the U2F protocol with physical tokens, like Yubikey. In the protocol, domain name is used in negotiating the handshake, so just slightly incorrect domain name disrupts it.

If a target organization monitors login events, they could notice numerous logins coming from a single IP address – the Evilginx2 server. In the case when this IP address is not in a geographical proximity to the victim, it could also trigger an alert. However, this is not a bulletproof protection, since an attacker could use a number of IP addresses or one of the VPN services to change their IP frequently and in close proximity to the victims. There are also many real-world situations where this would be a false alarm.

Creating an allow list of IP addresses, from which logging in is possible can mitigate this attack as well. If Evilginx2 server is not able to load the login page and log in, it cannot send its modified version to a victim.

More sensitive data and actions could be protected by another layer of authentication and require another login.

[22] https://github.com/elceef/dnstwist

CONCLUSION

In this whitepaper we showcased a phishing with MFA bypass. We described an architecture, which makes it possible and convenient to create and run phishing campaigns while combining multiple tools and using each's strengths.

This infrastructure uses three main components – Gophish for keeping list of the targets, email creation, sending the email campaigns and visualization, Evilginx2 as a landing page, which acts as a reverse proxy and impersonates legitimate login page and autonomous scripts, which can perform actions on behalf of the phished users in an automated manner, immediately after they can gain access.

Cooperation of these tools was achieved by their specific configuration and a few scripts. Gophish is configured to use unique links to Evilginx2, which are imported through Gophish API by a custom script of ours. For user tracking, Gophish tracking ids are exported to Evilginx2, which is modified to notify Gophish about any clicked links and submitted passwords. We also created a script, which reads session tokens from the Gophish database and transmits them into a JSON file to be read by autonomous scripts. We also modified both Gophish and Evilginx2 to not send the HTTP and SMTP data specific to them and made Evilginx2 log into files.

While multi-factor authentication is a useful tool for numerous reasons, we demonstrate it is not a cure-all for every phishing scenario and can be bypassed relatively easily, using only off-the-shelf tools. Companies should not rely solely on MFA to protect them against phishing. We also provide a number of possible mitigations which could help reveal or mitigate such attack.

LIFARS
your digital world, secured
244 Fifth Avenue, Suite 2035, New York, NY 10001
LIFARS.com (212) 222-7061 info@lifars.com