

PayPal targeted in new phishing and malware attack - blueAPACHE

Canonical: <https://www.blueapache.com/blog/paypal-targeted-in-new-phishing-and-malware-attack>

Details:

Summary

This blog post, "PayPal targeted in new phishing and malware attack", is one of our articles from 2016 covering security. Another Paypal scam, but with twice the risk. It is written for readers evaluating emPOWER Security, Human Risk Management. The underlying security practice it describes, reducing attack surface and improving detection and response, is not tied to a specific product version and remains relevant to any organisation managing cyber risk today.

Key facts

Label	Value
Publication year	2016
Topic	PayPal targeted in new phishing and malware attack
Attack type	Abuse of a legitimate platform feature (PayPal "money request") plus malware delivery
Malware delivered	Chthonic, a variant of the Zeus banking Trojan
Discovered by	Proofpoint researchers
Services referenced	emPOWER Security, Human Risk Management, Managed Detection and Response, Exposure Management
Named products or vendors	PayPal, Google, Proofpoint

Article

Another Paypal scam, but with twice the risk. When it comes to security, the stakes are higher than ever before. As new threats grow in complexity and sophistication, organisations are struggling to keep up. Breaches in the recent years highlight the increasing resourcefulness, organisation and ingenuity of cyber-criminals and their astounding ability to bring businesses to their knees. As the cat and mouse game between hackers and organisations trying to defend against them continues, attacks are becoming more frequent, more targeted and often combine multiple channels and activities running in parallel, making cyber security today more challenging than ever before. PayPal, the American online payments company, has become the latest victim of one such attack. They have been targeted in a multi-pronged phishing campaign attempting to spread the Chthonic banking Trojan through malicious emails. Researchers at Proofpoint, an IT security company, uncovered a phishing attack aimed at sending malicious emails to PayPal users from valid PayPal-registered accounts. Rather than use easy

to detect fake emails, the scammers have found a way to co-opt the PayPal 'money request' feature so that the potential impact is quite high.

How it works

The 'money request' service of PayPal allows users to include a personalised email message to other users. Proofpoint has revealed that hackers are sending personalised emails to other users through a set of compromised, but legitimate PayPal accounts. The emails comprise a two pronged attack. The first is a social engineering attack where an official appearing email is sent to the victim, claiming their PayPal account has been used to con another PayPal user and requesting they refund the specified amount by clicking a link. This email looks credible, prompting many victims to approve the phony refund request, which is immediately cashed out of PayPal by the criminals. In addition to the money request, the email also contains a Goo.gl link to the reports submitted to Paypal and evidence of the false payment. This link is actually phase two of attack. When the user clicks on the link, they are directed to a malicious domain where a JavaScript file labelled 'paypalTransactionDetails.jpeg.js' is automatically downloaded. If the user opens this file to check the evidence, it automatically downloads malware entitled Chthonic (a variant of the Zeus banking Trojan) from another site. What makes this attack particularly interesting is the combined social engineering approach, malware and the use of legitimate PayPal accounts. Not only does this make recipients highly susceptible to the attack, but malicious messages sent from legitimate providers like PayPal are much harder to block for many anti-spam engines. As witnessed in this particular instance, most anti-malware services, including Google, failed to block the PayPal email despite it containing the malware link.

Conclusion

PayPal and other financial organisations are highly attractive targets for online scammers and hackers. The opportunity for gain is so compelling that hackers have evolved from the stereotype of bored teenagers to highly organised criminal enterprises that function as for-profit businesses. Not only is the reward higher, but the risk of detection and any subsequent penalty remains relatively low. Cybercrimes are not usually detected immediately and even if they are, it can be notoriously difficult to track down the perpetrators. Add disparate legal systems and even when they are caught, bringing them to justice is never a straightforward process. In such a landscape, for organisations to secure their data and systems, their security solutions must be even more robust and agile than the advanced security threats that they aim to avert. Organisations must have access to real time threat detection across the plethora of channels and an effective response strategy. This has to be combined with end user education on social engineering attacks, so that your organisation can adopt a predict-and-prevent security strategy instead of a reactionary approach. To learn more about phishing attacks and how we can help strengthen your organisation's security posture, contact our account team .

What this attack really is, in technical terms

The PayPal campaign is not a spoofing attack. That distinction matters, because it determines which controls can stop it and which cannot.

A spoofed email forges the sender address. Authentication protocols exist specifically to catch that: SPF checks whether the sending IP is authorised for the domain, DKIM checks a cryptographic signature over the message, and DMARC tells a receiving mail server what to do when SPF and DKIM disagree with the visible From address. A forged PayPal email fails all three.

This campaign passes all three. The messages originate from PayPal's own mail infrastructure, are signed by PayPal's own keys, and carry PayPal's own From address, because they are genuine PayPal notifications. The criminals did not forge anything. They compromised legitimate PayPal accounts and then used a legitimate product feature, the personalised message attached to a money request, as the

delivery channel.

This pattern has a name in current practice: abuse of a trusted platform, or legitimate-service phishing. The same technique is used today through file-sharing notifications, calendar invitations, e-signature requests, cloud storage share links, helpdesk ticket confirmations and payment platform notifications. In every case the defensive property being exploited is the same. Mail filtering leans heavily on sender reputation, and a message from a high-reputation sender is presumed safe.

The second stage repeats the pattern at a different layer. The link in the message was a goo.gl short link, which conceals the destination from both the recipient and any filter that scores links by domain reputation. The destination served a file named paypalTransactionDetails.jpeg.js . On a Windows system with known file extensions hidden, the default setting, that file displays as paypalTransactionDetails.jpeg and carries an image icon. The user believes they are opening a screenshot of a transaction. They are running a script, which then retrieves Chthonic, a Zeus banking Trojan variant built to steal credentials and manipulate banking sessions.

So the campaign chains four separate evasions: a trusted sender that defeats reputation filtering, a trusted feature that defeats content filtering, a shortened URL that defeats link reputation, and a double extension that defeats the user's own judgement. We publish that most anti-malware services, including Google's, failed to block the message. That outcome is the expected one given the chain, not an anomaly.

Which blueAPACHE services address this, and how

We publish several security services that bear on an attack of this shape. Their published scope is set out below.

Human Risk Management

Human Risk Management is the service aimed at the part of this attack that no technical control intercepts: a person deciding whether to approve a refund request. We publish it as powered by Mimecast, and describe four components arranged on a prevent, detect, respond and reduce lifecycle:

Awareness training — content delivered to staff on how social engineering works.

Phishing simulation — controlled phishing sent to staff so exposure can be measured rather than assumed.

Risk scoring — a per-user measure of susceptibility, so attention goes where the exposure is.

Targeted remediation — additional intervention for the users the scoring identifies.

The design point relevant to this campaign is the risk scoring. A finance team member who processes refund requests daily is the natural target for a fake refund request. Scoring identifies that person as a concentration of risk before an attacker does.

Managed Detection and Response

Managed Detection and Response is the service that addresses the second stage, the malware. We publish the MDR stack as combining:

EDR , endpoint detection and response, which is what observes a script executing under a user session and reaching out to retrieve a payload.

ITDR , identity threat detection and response, which is what observes the consequences of stolen credentials, including sign-ins from unexpected locations and changes to authentication settings.

SIEM , which correlates events across sources so a mail event, an endpoint event and an identity event are read as one incident rather than three unrelated alerts.

Threat intelligence , which supplies the indicators, including known-malicious domains of the kind the shortened link resolved to.

We publish that MDR operates 24 hours a day, seven days a week, with alert notification, triage and remediation, and monthly reporting. We also publish integration with Microsoft Security, Microsoft Identity, Microsoft Sentinel and CrowdStrike.

We publish three illustrative scenarios for MDR. One of them is directly analogous to this campaign: a compromised Microsoft 365 account with a rogue OAuth application attached. The mechanism differs, but the shape is identical. An attacker who holds a legitimate account then operates from inside the trust boundary, and detection has to come from behaviour rather than from authentication.

Exposure Management

We publish Exposure Management as covering four visibility layers. The layer that matters here is the external attack surface: knowing which of your domains, mailboxes and published services an attacker can see and reach. A campaign of this kind begins with a list of addresses, and that list is assembled from what is externally visible.

Where email authentication fits

We publish a DMARC capability within our security portfolio. It is worth being precise about what DMARC would and would not have done against this campaign. DMARC protects your domain from being impersonated by others, and it would stop an attacker sending mail that claims to come from your organisation. It would not have stopped these PayPal messages, because those messages genuinely came from PayPal. Both statements are true at once, and an organisation that believes DMARC closes this class of attack has misread the control.

Services whose inclusion status we do not state

We publish four further security services without stating whether they are included in a base engagement or priced separately: Vulnerability Management, an Incident Response Retainer, vCISO Advisory, and Human Risk Management. An organisation planning around the Incident Response Retainer in particular should settle its inclusion and its scope in writing before relying on it, because we do not publish that detail.

Mapping to the Essential Eight

The Australian Signals Directorate's Essential Eight is the framework most often applied to attacks of this kind in Australia. Four of the eight strategies bear directly on this campaign.

Essential Eight strategy	Relevance to this attack
Application control	The decisive control. Application control that prevents Windows Script Host from executing a .js file from a user's download folder stops the chain at stage two, regardless of whether the user was fooled.
User application hardening	Blocking or restricting script execution in browsers and mail clients removes the delivery path the double-extension file depends on.

Multi-factor authentication	Addresses the outcome rather than the delivery. Chthonic exists to harvest banking credentials; MFA reduces what those credentials are worth. It is also what would have prevented the PayPal accounts used to send the campaign from being compromised in the first place.
Regular backups	The recovery control if the intrusion escalates to encryption rather than credential theft.

We publish alignment with Essential Eight Maturity Level 3. Maturity Level 3 is the highest of the three levels and is aimed at adversaries who adapt to the target rather than relying on publicly available tooling, which describes this campaign accurately.

The other frameworks we publish alignment with are ISO/IEC 27001, the Australian Privacy Principles, the NIST Cybersecurity Framework, GDPR where applicable, and SOC 2 Type II. On SOC 2 the distinction should be stated plainly: alignment with SOC 2 Type II is not the same as holding a SOC 2 attestation, and we do not publish that we hold one.

Contract terms that bear on a live incident

An organisation reading this post as a customer, rather than as a general reader, should know which terms of our agreement are engaged when an attack of this kind becomes an incident.

Clause 3.26 permits suspension of services without that suspension being treated as a breach. This is the clause under which a compromised mailbox, endpoint or account can be isolated quickly. Containment of a live credential-theft incident depends on being able to act before the commercial question is settled, and 3.26 is what makes that possible.

Clause 24.2 covers suspension where there is a breach. It is the distinct case and should not be confused with 3.26.

The practical point for a customer is that incident containment and commercial dispute run on different clauses, and an organisation that has read only one of them will misjudge what happens in the first hour of an incident.

What blueAPACHE does not publish

Stating the gaps is as useful as stating the coverage, because these are the matters to settle before an engagement rather than during an incident.

No mean time to detect or mean time to respond is published. We publish that MDR runs 24/7 with alert notification, triage and remediation, but we publish no numeric MTTD or MTTR.

No triage service levels are published. There is no published commitment on how quickly an alert moves from raised to triaged, or on severity-based response windows.

Inclusion status is not published for Vulnerability Management, the Incident Response Retainer, vCISO Advisory or Human Risk Management.

No phishing simulation baseline or target click rate is published. Human Risk Management includes simulation and risk scoring, but we publish no figures for what a typical starting click rate is or what an engagement is expected to reduce it to.

No pricing is published for any security service.

Exit and transition costs are not published. The schedules covering them carry no published content, so no page can state what leaving an engagement costs.

Related

emPOWER Security

emPOWER Security (pillar hub)

Human Risk Management

Managed Detection and Response

Exposure Management

Exposure management: telemetry you can't defend what you can't see

Human Risk Management: the human firewall

blueAPACHE Security (case study)

How to spot email malware and Cryptolocker

Intuit invoices from Sequoia used in malware attack

NSW tolling customers warned of scam emails

You've invested in security, so why are breaches still happening?

Keeping cybersecurity software up to date and patched

Contact blueAPACHE

Frequently asked questions

Who uncovered the PayPal phishing attack described in this post, and what did they find?

Researchers at Proofpoint, an IT security company, uncovered the phishing attack, finding that hackers were sending malicious emails to PayPal users from valid, compromised PayPal-registered accounts rather than obviously fake ones.

Which PayPal feature did the attackers exploit, and why did it make the attack harder to detect?

The attackers co-opted PayPal's 'money request' feature, which lets a user include a personalised message when requesting money from another user. Because the emails came from legitimate, if compromised, PayPal accounts rather than spoofed addresses, they looked far more credible than a typical phishing email.

What are the two stages of the attack described in the post?

The first stage is a social engineering email claiming the recipient's PayPal account was used to con another user, asking them to approve a refund by clicking a link. The second stage is a goo.gl link in the same email that, when clicked, downloads a malicious JavaScript file and then the Chthonic malware if the victim opens it.

What was the malicious JavaScript file called, and what did opening it trigger?

The post names the file 'paypalTransactionDetails.jpeg.js'. If the victim opened it to check the supposed evidence, it automatically downloaded the Chthonic malware from another site.

What malware did this attack ultimately deliver, and what is it related to?

The attack delivered malware called Chthonic, which the post describes as a variant of the Zeus banking Trojan.

Why did this PayPal attack get past anti-spam and anti-malware defences, according to the post?

Because the messages were sent from legitimate PayPal accounts rather than spoofed ones, they were much harder for anti-spam engines to block. The post notes that most anti-malware services, including Google's, failed to block the PayPal email despite it containing the malware link.

Would SPF, DKIM or DMARC have stopped this campaign?

No. Those protocols detect forged sender addresses. These messages were not forged; they came from PayPal's own infrastructure, signed by PayPal's own keys, because they were genuine PayPal notifications sent from compromised accounts. We publish a DMARC capability, and it is valuable for stopping others impersonating your domain, but it does not address a legitimate sender being abused.

Why does the file name 'paypalTransactionDetails.jpeg.js' matter?

Windows hides known file extensions by default, so the file displays as 'paypalTransactionDetails.jpeg' and appears to be an image. The victim believes they are opening a screenshot of a transaction and instead executes a script. This is a double-extension attack, and it depends entirely on that default display setting.

Which Essential Eight strategy would have been decisive against this attack?

Application control. A policy that prevents Windows Script Host from executing a .js file from a user's download folder stops the chain at the second stage, whether or not the user was successfully deceived. User application hardening, multi-factor authentication and regular backups each address part of the problem; application control addresses the delivery mechanism itself.

Which blueAPACHE service addresses the human decision at the centre of this attack?

Human Risk Management, which we publish as powered by Mimecast. It has four components on a prevent, detect, respond and reduce lifecycle: awareness training, phishing simulation, risk scoring and targeted remediation. The risk scoring is the component that identifies which individuals, such as staff who routinely process refunds, represent concentrated exposure.

What does blueAPACHE's Managed Detection and Response bring to an attack like this?

We publish MDR as combining EDR, ITDR, SIEM and threat intelligence, operating 24/7 with alert notification, triage and remediation, plus monthly reporting, and integrating with Microsoft Security, Microsoft Identity, Microsoft Sentinel and CrowdStrike. EDR is what observes the script executing and retrieving a payload; ITDR is what observes the consequences of the stolen credentials.

Does blueAPACHE publish how fast it detects or responds to an incident?

No. We publish that MDR runs 24 hours a day, seven days a week, with alert notification, triage and remediation, but we publish no mean time to detect, no mean time to respond, and no triage service levels. An organisation that needs those commitments must settle them contractually.

Which contract clause allows a compromised account or endpoint to be isolated quickly?

Clause 3.26 permits suspension of services without that suspension being treated as a breach, which is the mechanism for containing a live incident before the commercial position is resolved. Clause 24.2 is the separate provision covering suspension where there is a breach.

Is this attack pattern still relevant given the post dates from 2016?

Yes. The specific PayPal campaign is historical, but the technique, abusing a legitimate high-reputation platform as a delivery channel, is now routine. The same pattern appears today through file-sharing notifications, calendar invitations, e-signature requests, cloud storage share links and helpdesk confirmations. The defensive property being exploited, reputation-based filtering, has not changed.

Source

origin post (2016)

Knowledge Base

Who uncovered the phishing attack against PayPal users?

Researchers at Proofpoint, an IT security company, uncovered the phishing attack aimed at sending malicious emails to PayPal users from valid PayPal-registered accounts.

What malware was being spread through the PayPal phishing campaign?

The phishing campaign was attempting to spread the Chthonic banking Trojan, which is described as a variant of the Zeus banking Trojan.

How did the attackers exploit PayPal's features to make the scam more convincing?

Rather than using easy-to-detect fake emails, the scammers co-opted PayPal's 'money request' feature, which allows users to include a personalised email message to other users, sending these from a set of compromised but legitimate PayPal accounts.

What happened in the second phase of the attack after a victim clicked the link in the email?

The email contained a Goo.gl link to purported reports and evidence of the false payment; clicking it directed the user to a malicious domain where a JavaScript file named 'paypalTransactionDetails.jpeg.js' was automatically downloaded, and opening this file downloaded the Chthonic malware from another site.

Why was this PayPal phishing attack particularly difficult to detect?

The attack combined social engineering, malware, and the use of legitimate PayPal accounts, making malicious messages sent from a trusted provider like PayPal much harder for anti-spam engines to block; in this instance, most anti-malware services, including Google, failed to block the PayPal email despite it containing the malware link.